

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

## **IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

# Knobbe Martens Olson & Bear LLP

## Intellectual Property Law

2040 Main Street  
Fourteenth Floor  
Irvine, CA 92614  
Tel 949-760-0404  
Fax 949-760-9502  
knob.com

### ORIGINAL WILL FOLLOW VIA:

- ☐ MAIL  
☐ INTERNATIONAL AIRMAIL  
☐ COURIER  
☒ WILL NOT FOLLOW  
☐ HAND DELIVERY  
☐ WITH ENCLOSURES  
☐ WITHOUT ENCLOSURES

## Facsimile Transmittal Sheet

### Confidentiality Notice:

Don W. Martens  
James B. Bear  
Darrell L. Olson  
William B. Bunker  
William H. Nicman  
Arthur S. Rose  
James F. Lesniak  
Ned A. Israelson  
Drew S. Hamilton  
Jerry T. Sewell  
John B. Spangola, Jr.  
Edward A. Schlatter  
Gerard von Hoffmann  
Joseph R. Ro  
Catherine J. Holland  
John M. Carlson  
Karen Vogel Weil  
Andrew H. Simpson  
Jeffrey L. Van Hooser  
Daniel E. Altman  
Stephen C. Jenson  
Vito A. Canuso III  
William H. Shreve  
Lynda J. Zadra-Symest  
Steven J. Natausky  
Paul A. Stewart  
Joseph F. Jennings  
Craig S. Summers  
AnneMarie Kaiser  
Brandon R. Baeck  
Thomas F. Smegal, Jr.  
Michael H. Trenholm  
Diane M. Reed  
Ronald J. Schoonbaum  
John R. King  
Frederick S. Baratta  
Nancy W. Vensko  
John P. Giezantanner  
Adele S. Akhtar  
Thomas R. Arno  
David N. Weiss  
Daniel Hart, Ph.D.  
Douglas G. Muonihauzer  
Lori Lee Yamato  
Michael K. Friedland  
Dale C. Hunt, Ph.D.  
Ricardo E. Campbell  
Stacey R. Halpern  
Lee W. Henderson, Ph.D.  
Mark M. Abumori  
Jon W. Gurka

John W. Holcomb  
Joseph M. Reisman, Ph.D.  
Michael L. Fuller  
Eric M. Nelson  
Mark R. Benedict, Ph.D.  
Paul N. Conover  
Robert J. Roby  
Sabing H. Lee  
Kathleen A. Delaney  
Joseph S. Cianfrani  
William R. Zimmerman  
Paul C. Steinhardt  
Eric S. Furman, Ph.D.  
Susan M. Natland  
James W. Hill, M.D.  
Deborah S. Shepherd  
Glen L. Nuttall  
Tirzah Abd Lowe  
Sanjivpal S. Gill  
Rose M. Thieszen, Ph.D.  
Michael A. Guillana  
Mark J. Kertz  
Rabinder N. Narula  
Bruce S. Itchkawitz, Ph.D.  
Michael S. Okamoto  
John M. Grover  
Mollary K. De Merit  
Iran A. Latof  
Amy C. Christensen  
Sharon S. Ng  
Mark J. Gallagher, Ph.D.  
David G. Jankowski, Ph.D.  
Brian C. Horra  
Payson J. LeMouleur  
Shella N. Swarcep  
Ben A. Katzenellenbogen  
Linda H. Liu  
Andrew N. Merickol, Ph.D.  
David L. Hauser  
James F. Herkenhoff  
Scott Loras Murray  
Andrew M. Douglas  
Marc T. Morley  
Salima A. Marani, Ph.D.  
Sam K. Tahmassebi, Ph.D.  
Christy G. Lee  
Jonathan A. Hyman  
Curtis C. Doslor  
Joseph J. Mallon, Ph.D.  
Thomas P. Krzeminski  
Sean M. Murray

Eleanor Niu  
Valerie L. Bracken  
J. David Evered  
Perry D. Oldham  
Jerry L. Hefner, Ph.D.  
Russell M. Jolde  
Abraham W. Chuang  
Pui Tong Ho, Ph.D.  
Erik T. Anderson  
John L. Paik  
Jesse A. Rothwell  
Marc C. Baumgartner  
Daniella Klausner  
Kyle F. Schuster  
Raphael A. Gutierrez  
Nathan A. Engels  
Gregory A. Hormanson  
Zi Y. Wong  
John N. Karsara  
Matthew S. Bellingier  
David K. Wiggins  
Darryl H. Steensma, Ph.D.  
Lauren J. Kohler  
Ted M. Cannon  
Carol M. Pitzel  
Josue A. Villalva  
Shella R. Gibson  
Andrew I. Kimmel  
Curtis R. Huffmire  
Tina Chen  
Brandon Gingrich, Ph.D.  
Christopher L. Ross  
Don W. Anthony, Ph.D.  
John G. Rickenbrose  
Aaron D. Barker  
Christian A. Fox  
M. Todd Hales  
Ell A. Louts, Ph.D.  
Jennifer L. Enmon, Ph.D.  
Ryan E. Melnick, Ph.D.  
Brian C. Leubitz  
Yanna S. Bouris  
Philip M. Nelson  
Karl L. Klassen  
Jason T. Evans  
Walter S. Wu  
Julie M. McCloskey  
Katsuhiko Arai  
C. Philip Poirier  
Mark A. Geier

Of Counsel  
Louis J. Knobbe  
Jerry R. Seller

Japanese Patent Atty  
Katsuhiko Arai  
Tomohisa Sugiyama

Korean Patent Atty  
Mincheol Kim  
Heungsoo Choi

Scientists & Engineers  
(Non-Lawyers)

Raimond J. Saloniak  
Khuram Rahman, Ph.D.  
Jennifer Haynes, Ph.D.  
Cha S. Chereskin, Ph.D.  
James W. Ausley  
Connie C. Tong, Ph.D.  
Suzanne Jackson, Ph.D.  
Nira M. Brand  
Tiffany C. Miller  
James W. Chang, Ph.D.  
Marina L. Gordoy, Ph.D.  
W. Frank Dauerer  
Lang J. McHardy  
Karen J. Lonkar  
Chris Westberg, Ph.D.  
Eric B. Ives, Ph.D.  
Raymond D. Smith, Ph.D.

\* A Professional Corporation  
† Also Barister At Law (Hong Kong)  
\*\* U.S. Patent Agent  
†† Also Solicitor (China & Taiwan)

The documents accompanying this facsimile transmission contain confidential information which may be legally privileged. The information is intended only for the use of the recipient named below. If you have received this facsimile in error, please immediately notify us by telephone to arrange for return of the original documents to us. Any disclosure, copying, distribution or the taking of any action in reliance on the contents of this faxed information is strictly prohibited.

TO: Examiner Kuen S. Lu, Group Art Unit 2177  
FIRM: UNITED STATES PATENT AND TRADEMARK OFFICE  
FACSIMILE NO: 703-746-9694  
YOUR REFERENCE: U.S. Patent Application No. 10/050,675 OUR REFERENCE: FNDSTN.013A  
FROM: Ted M. Cannon, Reg. No. 55,036  
TOTAL NUMBER OF PAGES: 104 (INCLUDING COVER SHEET). **PART 1 OF 2**  
OPERATOR: Kristin Eldred DATE: July 1, 2004 TIME:  
IF YOU DID NOT RECEIVE ALL OF THE PAGES, PLEASE CALL 949-721-2849 IMMEDIATELY.  
MESSAGE: Dear Examiner Lu:

In accordance with our telephone conference. I am attaching copies of the documents you requested. Please call Jerry T. Sewell at 949-721-2849 or Kristin Eldred at 949-721-2846 if you need additional information.

JTS-20289.DOC:ke  
20040630

550 West C Street  
Suite 1200  
San Diego CA 92101  
Tel 619 591-0000

201 California Street  
Suite 1450  
San Francisco CA 94111  
Tel 415 774-4444

1901 Avenue of the Stars  
Suite 1500  
Los Angeles CA 90067  
Tel 310 554-9400

3403 Tenth Street  
Suite 700  
Riverside CA 92501  
Tel 951 507-0774

1114 Marsh Street  
San Luis Obispo CA 93401  
Tel 805-547-5580  
Fax 805-547-5580

1 Steven Brower, State Bar #93568  
2 Brian Barrow, State Bar #177906  
3 **STEPHAN, ORINGHER RICHMAN & THEODORA**  
4 A Professional Corporation  
5 535 Anton Boulevard, Suite 800  
6 Costa Mesa, California 92626  
7 Telephone: (714) 241-0420

8 Attorneys for Defendants

9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
**SUPERIOR COURT OF THE STATE OF CALIFORNIA**  
**FOR THE COUNTY OF ORANGE**

FOUNDSTONE, INC.,

Plaintiff,

vs.

NT OBJECTIVES, INC., et al.,

Defendants.

CASE NO: 02CC15350  
Judge David H. Brickner  
Dept. C17

**DECLARATION OF DAN  
KUYKENDALL IN OPPOSITION TO  
OSC RE: PRELIMINARY  
INJUNCTION**

Date: October 25, 2002  
Time: 3:00 p.m.  
Dept: C17

I, Dan Kuykendall, declare:

1. The facts stated herein are of my own personal knowledge except for those set forth on information and belief, and as to any such facts, I have a basis for believing that they are true. If called upon to testify as a witness I could and would competently do so as set forth herein.

**ROLE IN THIS LITIGATION**

2. I am individually named as a defendant in this matter.

3. I was employed by Plaintiff Foundstone, Inc. from August 2001 until July 1, 2002, as more fully set forth below. I was never an officer or director of that company.

D:\wpdocs\sb102211.02

1

**DECLARATION OF DAN KUYKENDALL IN OPPOSITION TO OSC RE: PI**

1 11. Before I started on a new project at Foundstone, I was usually given a list of the  
2 desired features. I was not told what algorithms to use or how to make those features actually come  
3 into existence in the software. I was expected to figure out, based on my prior knowledge and  
4 experience, how to make the computer program do what was expected.

5 12. One of the features on which I worked was the FoundScan web portal (aka  
6 Experience). When I started I wrote the web interface to scan management in about three weeks.  
7 It was a good application, but there was nothing I did which was not the implementation of  
8 generally known computer programming practices. Moreover, the NTO toolkit does not have any  
9 web portal.

10 13. After that I helped with various tasks related to fixing and/or upgrading other parts  
11 of software programs at Foundstone. This included the creation of VulnTrak, a program which  
12 maintains records of vulnerabilities reported in client computer systems. It took me about a month  
13 to write the VulnTrak computer program. Once again, while it is a good implementation, there are  
14 no algorithms or methods used in that computer program which, in my experience, are not already  
15 generally known to good programmers with experience in database programming.

16 14. I was never asked to participate in the preparation of any application for a patent  
17 while I was at Foundstone. I have never been advised that I am named as the "inventor" on any  
18 patent application filed by Foundstone.

19 **FOUNDSTONE'S ALLEGED TRADE SECRETS**

20 15. I have read the Declaration of Stuart McClure submitted in support of this OSC re:  
21 Preliminary Injunction. From review of that document I am unable to determine any specific  
22 method, algorithm or technique which is being referred to by Foundstone as a trade secret, with the  
23 possible exception of Foundscore. However, NTO isn't doing anything like Foundscore.

24 16. As stated above, the work which I did while I was employed by Foundstone was good  
25 programming, but the methods, techniques and algorithms which I used were the same as those I  
26 would expect would be used by any other excellent experienced programmer.

27 17. The McClure declaration gives the impression that Foundstone might be claiming  
28 HTML as a trade secret. The pages which display on the world wide web are mostly HTML. All

D:\wpdocs\sb102211.02

3

DECLARATION OF DAN KUYKENDALL IN OPPOSITION TO OSC RE: PI

1 of the HTML techniques that I used at Foundstone were standard. No new ground was broken  
2 there. All of the HTML I have done at NTO is, likewise, standard programming techniques.

3 18. During the entire time I was employed by Foundstone I never saw any documents,  
4 source code or other material with any "trade secret" or "secret" designation.

5 DECISION TO LEAVE FOUNDSTONE AND JOIN NTO

6 19. My commute to the Foundstone office was too far. When I learned that my wife and  
7 I were going to have a baby I wanted to be closer to home so that I could assist my wife. I had told  
8 Foundstone, even during my initial interview, that I wanted to telecommute. However, I was only  
9 able to maintain the right to telecommute two days a week by threatening to quit. This made me  
10 uncomfortable with my position at Foundstone. Moreover, I found the pace of work at Foundstone  
11 to be frustrating. I would have little to do for an extended period of time while management created  
12 a document stating what features they wanted in the software. Then I would need to work  
13 frantically, for a couple of months, actually creating the program.

14 20. I had no contact with JD, or anyone else at NTO, regarding potential employment,  
15 until after I left Foundstone. I gave my two week notice in mid-June. Our baby was born on  
16 June 27, 2002 and my last day at Foundstone was July 1, 2002.

17 21. After I left my employment at Foundstone I started to look for other employment. For  
18 a period of time I tried to work from my home but I decided I needed a more steady income. I  
19 contacted JD who I knew had started NTO after he left Foundstone. He told me that he was  
20 interested but that I would need to wait until he had enough money to fund a position for me.

21 22. I subsequently joined NTO where I am presently the Senior Internet Software  
22 Engineer.

23 23. We have, as a result of this litigation, been required to spend time meeting with our  
24 legal counsel. Based on those discussions, and based on my discussions with the other employees  
25 of NTO, and based on my knowledge of computer programming and computer techniques, neither  
26 I, nor any other person at NTO, is utilizing anything which I understand to constitute a trade secret  
27 of Foundstone in the furtherance of the business of NTO.

28

D:\wpdocs\sb102211.02

4

DECLARATION OF DAN KUYKENDALL IN OPPOSITION TO OSC RE: PI

1 Steven Brower, State Bar #93568  
2 Brian Barrow, State Bar #177906  
3 **STEPHAN, ORINGER RICHMAN & THEODORA**  
4 A Professional Corporation  
5 535 Anton Boulevard, Suite 800  
6 Costa Mesa, California 92626  
7 Telephone: (714) 241-0420  
8  
9 Attorneys for Defendants  
10

11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
**SUPERIOR COURT OF THE STATE OF CALIFORNIA**  
**FOR THE COUNTY OF ORANGE**

11 FOUNDSTONE, INC.,

12 Plaintiff,

13 vs.

14 NT OBJECTIVES, INC., et al.,

15 Defendants.  
16  
17

CASE NO: 02CC15350  
Judge David H. Brickner  
Dept. C17

**DECLARATION OF MICHAEL J.  
MORTON IN OPPOSITION TO OSC  
RE: PRELIMINARY INJUNCTION**

Date: October 25, 2002  
Time: 3:00 p.m.  
Dept: C17

18 I, Michael J. Morton, declare:

19 1. The facts stated herein are of my own personal knowledge except for those set forth  
20 on information and belief, and as to any such facts, I have a basis for believing that they are true.  
21 If called upon to testify as a witness I could and would competently do so as set forth herein.

22 **ROLE IN THIS LITIGATION**

23 2. I am individually named as a defendant in this matter.

24 3. I was employed by Plaintiff Foundstone, Inc. from January 2001 until June 2002, as  
25 more fully set forth below. I was never an officer or director of that company.  
26  
27  
28

D:\wpdocs\sb102210.02

1

**DECLARATION OF MICHAEL J. MORTON IN OPPOSITION TO OSC RE: PI**

## BACKGROUND IN COMPUTERS

4. I received a bachelors degree (Phi Beta Kappa) in Information and Computer Science from the University of California at Irvine in June, 1994.

5. I have been involved with computers since 1982. Prior to entering college I was fluent with several computer assembly languages (6502, Z-80, 8088, 6809) and with microcomputer architecture. I had written computer programs that dealt with: graphics and realtime video chip programming; hardware interrupts; an 80486 assembler; dabbled in TTL/CMOS logic design; designed several circuits.

6. Since before college, and throughout my career, I have been evolving a program that does ray traced graphical rendering. This is a 3-D technique where you "draw" a ray of light with sufficient detail so that the final image has shadows, shading, perspective and reflections, all generated automatically. Part of the program includes the creation and implementation of a "description language," which is similar to creating a new computer programming language and a new computer program which understands that language and causes the computer to do what has been requested.

7. I worked for Quarterdeck Corporation from June 1994 to October 1996 as a staff programmer. I learned about Windows programming and TCP/IP programming. During this employment I wrote a PPP layer to the Winsock product, which is a networking protocol related to Windows.

8. I worked for Connect3 from October 1996 to April 1997 as a software engineer. I learned database programming including ODBC and SQL.

9. I worked for Beckman (now Beckman/Coulter) from April, 1997 to February, 2000 as a senior software engineer. I did more database programming (ODBC, Oracle and SQL Server), more networking (TCP/IP) and user interface Windows programming. I learned COM and ATL. All of this was done for a DNA Sequencer product referred to as the CEq-2000.

10. I worked for HiHo.com from February 2000 to December 2000 as a senior software engineer. I learned HTML, DHTML, ASP, WinInet, MTS and ActiveX, among others. The application we were working on was a distributed enterprise web application with a SQLServer

D:\wpdocs\sb102210.02

2

DECLARATION OF MICHAEL J. MORTON IN OPPOSITION TO OSC RE: P1

1 database. This job, in my mind, represented a culmination of my prior experience with various  
2 computer technologies and added several new ones to my repertoire (all having to do with web  
3 programming.)

#### 4 EMPLOYMENT BY FOUNDSTONE

5 11. I was happy with my employment at HiHo. However, it was a casualty of the dot com  
6 decline. On the day that it became apparent to me that HiHo was dissolving, I received a telephone  
7 call from Bernard Lee, a recruiter, who pointed me to Foundstone.

8 12. After a lengthy background check I began to work for Foundstone in mid-January  
9 2001. I was a senior software engineer for the first 3 months then chief software engineer after that.

10 13. Before I started on a new project at Foundstone, I was usually given a list of the  
11 desired features. I was not told what algorithms to use or how to make those features actually come  
12 into existence in the software. I was expected to figure out, based on my prior knowledge and  
13 experience, how to make the computer program do what was expected.

14 14. While employed by Foundstone I designed and implemented a reporting mechanism  
15 which I am informed and believe they are still using. Foundstone had previously obtained this  
16 functionality by using a software program from an outside vendor, but Foundstone wanted to save  
17 money. Substantial portions of the software I wrote for Foundstone were based upon work I had  
18 done while working for Beckman. All of the graphics techniques I used in the reporting mechanism  
19 were previously present in my raytracer software (see ¶ 6). Although I showed the product to other  
20 people in the company and discussed features and functions with them, the design of the algorithms  
21 and the coding of the software was done solely by me, or by people working to implement my  
22 instructions. While the reporting mechanism is a good computer program, and while it provides the  
23 functionality to support that piece of Foundstone's business, I am unaware of any feature in that  
24 software package that is exclusive to Foundstone. Moreover, the Fire & Water software, to be  
25 released by NTO, does not include the reporting mechanism software which was created while I was  
26 at Foundstone.

27 15. While employed by Foundstone I designed and implemented the FASL (Foundstone  
28 Attack Scripting Language) that I am informed and believe they are also still using as part of their

D:\wpdocs\sb102210.02



1 vulnerability assessment. I had previous experience writing computer languages such as the 80486  
2 assembler which I had written before I worked at Foundstone. The graphics capabilities in that  
3 software were substantially based on my raytracer software (see ¶ 6). Although I showed the  
4 product to other people in the company and discussed features and functions with them, the design  
5 of the algorithms and the coding of the software was done solely by me, or by people working to  
6 implement my instructions. While it is a good software package, and while it provides the  
7 functionality to support that piece of Foundstone's business, I am unaware of any feature in that  
8 software package that is exclusive to Foundstone. Moreover, the Fire & Water software, to be  
9 released by NTO, does not include anything like FASL.

#### 10 FOUNDSTONE'S ALLEGED TRADE SECRETS

11 16. I have read the Declaration of Stuart McClure submitted in support of this OSC re:  
12 Preliminary Injunction. From review of that document I am unable to determine any specific  
13 method, algorithm or technique which is being referred to by Foundstone as a trade secret.

14 17. To the extent that I am generally familiar with the graphical techniques which exist  
15 in the Foundstone software, because they exist in software I worked on while I was at Foundstone,  
16 it is my testimony that all of the graphical techniques that Foundstone may be claiming as  
17 proprietary are documented in public domain materials. Specifically, to my recollection, they can  
18 all be found in a book called "Computer Graphics Principles and Practice" by Foley, VanDam,  
19 Feiner and Hughes.

20 18. To the extent that I am generally familiar with the mathematical techniques which  
21 exist in the Foundstone software, because they exist in software I worked on while I was at  
22 Foundstone, it is my testimony that all of the mathematical techniques that Foundstone may be  
23 claiming as proprietary are documented in public domain materials, such as a standard linear algebra  
24 book. The cubic spline interpolation was done by copying code samples in a book called  
25 "Numerical Recipes" by William H. Press, Brian P. Flannery, Saul A. Teukolsky and William T.  
26 Vetterling.

27 19. The McClure declaration gives the impression that Foundstone might be claiming  
28 HTML as a trade secret. The pages which display on the world wide web are mostly HTML. All

D:\wpdocs\sb102210.02

1 of the HTML techniques that I used at Foundstone were standard. No new ground was broken there.  
2 All of the HTML I have done at NTO is, likewise, standard programming techniques.

3 **DECISION TO LEAVE FOUNDSTONE AND JOIN NTO**

4 20. JD Glaser and I became friends while we were both working at Foundstone. Prior to  
5 giving notice at Foundstone I asked JD whether I could work for his company. He advised me that  
6 under his Employment Agreement with Foundstone he could not have any such discussion with me  
7 while I was working for Foundstone.

8 21. Therefore, I gave my two week notice to Foundstone. My last day with Foundstone  
9 was June 28, 2002.

10 22. I subsequently applied for a job with NTO. I am presently the Senior Software  
11 Engineer for NTO.

12 23. We have, as a result of this litigation, been required to spend time meeting with our  
13 legal counsel. Based on those discussions, and based on my discussions with the other employees  
14 of NTO, and based on my knowledge of computer programming and computer techniques, neither  
15 I, nor any other person at NTO, is utilizing anything which I understand to constitute a trade secret  
16 of Foundstone in the furtherance of the business of NTO.

17 I declare, under penalty of perjury, that the foregoing is true and correct and that this  
18 declaration is executed on October 23, 2002, under the laws of the State of California.

19   
20 Michael J. Morton

1 Steven Brower, State Bar #93568  
Brian Barrow, State Bar #177906  
2 **STEPHAN, ORINGER RICHMAN & THEODORA**  
A Professional Corporation  
3 535 Anton Boulevard, Suite 800  
Costa Mesa, California 92626  
4 Telephone: (714) 241-0420  
5 Attorneys for Defendants  
6  
7

8 **SUPERIOR COURT OF THE STATE OF CALIFORNIA**  
9 **FOR THE COUNTY OF ORANGE**  
10

11 **FOUNDSTONE, INC.,**

12 Plaintiff,

13 vs.

14 **NT OBJECTIVES, INC., et al.,**

15 Defendants.  
16  
17

CASE NO: 02CC15350  
Judge David H. Brickner  
Dept. C17

**DECLARATION OF ERIK CASO IN  
OPPOSITION TO OSC RE:  
PRELIMINARY INJUNCTION**

Date: October 25, 2002  
Time: 3:00 p.m.  
Dept: C17

18 I, Erik Caso, declare:

19 1. The facts stated herein are of my own personal knowledge except for those set forth  
20 on information and belief, and as to any such facts, I have a basis for believing that they are true.  
21 If called upon to testify as a witness I could and would competently do so as set forth herein.

22 **ROLE IN THIS LITIGATION**

23 2. I am individually named as a defendant in this matter.  
24 3. I was employed by Plaintiff Foundstone, Inc. from May 2001 until June 2002, as more  
25 fully set forth below. I was never an officer or director of that company.  
26  
27  
28

D:\wpdocs\sh102212.02

1

DECLARATION OF ERIK CASO IN OPPOSITION TO OSC RE: PI

**BACKGROUND IN COMPUTER BUSINESS**

4. I obtained my bachelors degree in Business from California Polytechnic State University, San Luis Obispo in June, 1997. My coursework included web design/authoring and using complex modeling software to create statistical forecasting models.

5. From April 1998 to March 2000, I was employed by The Boeing Company as a Business Analyst, using software for financial modeling of return on investment and other business planning.

6. From March 2000 to February 2001, I was employed by Epoch Internet as a Product Manager. I designed software based products for a large Internet Service Provider (ISP) including email system and real-time provisioning systems for an online portal.

**EMPLOYMENT BY FOUNDSTONE**

7. A colleague of mine from a prior job referred me to Foundstone. I liked the idea of working for a small company which was known for doing good work.

8. I began to work for Foundstone on or about May 15, 2001. My title was Product Manager. My supervisor was Dave Cole, although I primarily reported to Stuart McClure.

9. I was involved in the design and execution of the FoundScan technology. That is, I worked on deciding what features should be included in the Foundscan software in order to provide the features which Foundstone clients either requested or which Foundstone believed would increase customer demand for Foundstone services. I was also responsible for Foundstone's beta testing software program, sales assistance, development of marketing related materials, ensuring Quality Assurance for software, and overall management of FoundScan related objectives.

10. My job responsibilities at Foundstone did not include any computer programming. Although I have worked with computer companies I have not actually done any computer programming.

11. I was never asked to participate in the preparation of any application for a patent while I was at Foundstone. I have never been advised that I am named as the "inventor" on any patent application filed by Foundstone.

**DIFFERENTIATING NTO AND FOUNDSTONE**

D:\wpdocs\sb102212.02

2

DECLARATION OF ERIK CASO IN OPPOSITION TO OSC RE: PI

12. As the person responsible for the overall business strategy of NTO I have been asked to explain the difference between Foundstone and NTO. Attached hereto as Exhibit "A," and incorporated herein by reference, is a document I prepared that provides a detailed description of the features and functions which will be included in the "Fire & Water" tools which NTO intends to distribute. Attached hereto as Exhibit "B," and incorporated herein by reference, is a document I prepared that provides my description of Foundscan, the software package utilized by plaintiff. As will be readily apparent, it is materially different from anything being done by the "Fire & Water" tool kit. Attached hereto as Exhibit "C," and incorporated herein by reference, is a comparison chart which I prepared showing features of FoundScan in comparison with those in the "Fire & Water" package. That document also shows features that are present in other software programs produced by entities who are not parties to this litigation.

#### FOUNDSTONE'S ALLEGED TRADE SECRETS

13. I have read the Declaration of Stuart McClure submitted in support of this OSC re: Preliminary Injunction. From review of that document I am unable to determine any specific method, algorithm or technique which is being referred to by Foundstone as a trade secret.

14. Although they are not clearly identified by the McClure Declaration, there are two databases which I believe might constitute the type of proprietary information to which Stuart McClure intends to refer. One of those might be referred to as the OS Fingerprint file. This would be the collection of information which is used by Foundstone to do Operating System identification. While I am informed that the technique of OS identification is not proprietary to Foundstone, their database of this information is extensive and might normally qualify as proprietary. The second would be the database of vulnerabilities. I am told that such a listing is actually a matter of public knowledge, but I believe this is something to which Stuart McClure was referring in this Declaration.

15. These two databases (OS Fingerprint and Vulnerabilities) are part of the FoundScan suite of security programs. During the time that I was employed by Foundstone, the OS Fingerprint database was maintained in an "unencrypted" form. That is, anyone who had access to the computer program could read and/or print the entire contents of the OS Fingerprint database.

D:\wpdocs\sb102212.02

1 16. As a regular part of my job at Foundstone, I was involved in sending "evaluation  
2 copies" of our software to the trade press so that they would review our products and provide us with  
3 coverage. When dealing with the press there are never any confidentiality agreements requested or  
4 obtained because the very purpose of sending the copies to the press is so that they will make public  
5 comments about your services and software.

6 17. I specifically recall that copies of the Foundstone software (and, therefore, full ability  
7 to view and/or print the contents of the databases) was provided to: John Taschek and Jim Rapoza  
8 of eWeek Labs; Andrew Conry-Murray of Network Magazine; Jane Parkhouse of SC Magazine;  
9 Mandy Andress of InforWorld; and, Konstantinos Karagiannis of PC Magazine. Each of those  
10 people, and their publications, had full and unfettered ability to view and/or print the entire contents  
11 of the Foundstone OS Fingerprint database.

12 18. Notwithstanding the potential public disclosure of this database, I want to be sure it  
13 is clear that NTO is not including any OS Identification in the "Fire & Water" package. Moreover,  
14 NTO does not have any copy of the Foundscan OS Identification database.

15 **DECISION TO LEAVE FOUNDSTONE AND JOIN NTO**

16 19. I left Foundstone because of its lack of interest in employee satisfaction and poor  
17 management. When I advised Dave Cole and Stuart McClure that I was quitting they asked if there  
18 was any job at the company, which they could give me, which would make me stay. When I turned  
19 them down they told me that I could return to Foundstone at any time in the future.

20 20. I began to look for a new job about a month before I left Foundstone. This was about  
21 a month after JD Glaser left Foundstone. I contacted him to see what he was doing since we had  
22 been friends while he was working for Foundstone. As a business person, rather than a technical  
23 person, I told him that I had some ideas about how to structure a successful organization.

24 21. I subsequently joined NTO where I am presently the Director, Product Strategy. My  
25 role is to develop and implement our overall product and business strategy.

26 22. We have, as a result of this litigation, been required to spend time meeting with our  
27 legal counsel. Based on those discussions, and based on my discussions with the other employees  
28 of NTO, and based on my knowledge of computer programming and computer techniques, neither

D:\wpdocs\sb102212.02

4

DECLARATION OF ERIK CASO IN OPPOSITION TO OSC RE: PI

1 I, nor any other person at NTO, is utilizing anything which I understand to constitute a trade secret  
2 of Foundstone in the furtherance of the business of NTO.

3 I declare, under penalty of perjury, that the foregoing is true and correct and that this  
4 declaration is executed on October 23, 2002, under the laws of the State of California.

5   
6 Erik Caso  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

D:\wpdocs\sb102212.02

5

DECLARATION OF ERIK CASO IN OPPOSITION TO OSC RE: PI

## **EXHIBIT 'B'**



**What is FoundScan?**

FoundScan is an advanced, network based "Vulnerability Management Technology" that is designed to identify all network based assets, their corresponding vulnerabilities, and provide a comprehensive system in which to manage the elimination of those vulnerabilities. FoundScan is deployed in a single machine or multi-machine architecture, and may be run as a distributed system where various servers are in separate locations. Deployment requires a FoundScan Consultant to be onsite performing the installation and setup, ranging from one to two days work. All FoundScan data is kept in a Microsoft SQL database, and requires the installation of this and other third party software in order to operate (the web portal requires Microsoft IIS to operate). All machines in a FoundScan system must be properly configured and maintained in order to operate properly; there is extensive work that must be performed and maintained in order to ensure proper communication between the machines in a FoundScan system or they will fail to work entirely.

Components and features of the "FoundScan Enterprise Vulnerability Management System":

- Discovery Engine - identifies, enumerates and classifies each host on a network. Classification can include the operating system, open ports, banners, hostname, and even device type (i.e. routers, firewalls, servers and 802.11 wireless devices.)
- Vulnerability Engine - includes the vulnerability checks that are performed by FoundScan and a database of all check related information (i.e. descriptions, recommendations, CVE numbers and more). FoundScan currently checks for more than 530 vulnerabilities, and grows in number on a weekly basis. The checks are performed using a scripting language developed in-house, called FASL (FoundScan Attack Scripting Language).
- Web Portal - a comprehensive, interactive web portal that allows for complete management of the FoundScan features. Governed by rule-based access, each user has an account with discreet permissions as to how they may interact with the system; account types include Administrator, Full and View Users. The portal allows for account creation/management, viewing of scan reports, data search functions, and remediation management control.
- Remediation Engine - embedded in the Web Portal, the remediation management component, dubbed VulnTrak, allows for work flow management of vulnerabilities discovered by FoundScan. Akin to a trouble ticket system, VulnTrak allows for an Administrator to assign vulnerabilities to users, who are then alerted to the ticket and must go and fix the security issue. Once completed, they may verify and close the ticket, returning it to the Administrator for archiving.
- Numerous ancillary features, including email alerting, iDefense security reports (provided by a third party), and much more.

FoundScan is designed, marketed and sold as an enterprise security solution used to identify vulnerabilities in a corporate network. It is not currently designed to handle numerous, non-recurring projects related to network management, as the investment in such work is much greater than the resulting output (i.e. an administrator that needs to create and run numerous discreet network tests would not use FoundScan due to the amount of time it takes to log into the system, create the proper scan(s), run them, log into a web portal account, select the

scan, view the information and then act.) FoundScan is designed to run as a network service, where an administrator creates recurring assessment scans, or jobs, which are run on a schedule. These jobs are then reviewed on a recurring basis and used to determine the security posture of an organization or network. Each job is configured to be automatically distributed to unlimited individuals, based on whether they have access to the job or not. FoundScan may be purchased as a licensed technology or a managed service.

**Feature comparison for Fire & Water and FoundScan**

The above general descriptions of each product attempt to clearly indicate the intended design of each product. With that considered, there are several features and capabilities common to both technologies, as well as many other competing technologies offered by other software vendors. While these features/capabilities may be shared, it is important to note that in not a single case is there a shared feature that is a proprietary technology of either company; various implementations and variations of these features/capabilities may be readily found in numerous software products and have been discussed and available in public forums long before the founding of either Foundstone (founded late 1999) or NT OBJECTives (founded mid-1997).

For a feature breakdown please see the Fire & Water - FoundScan Feature Comparison document.

## **EXHIBIT 'C'**

**FIRE & WATER FOUNDSCAN FEATURE COMPARISON\***

| FEATURE  | FIRE&WATER | FOUNDSCAN | EXISTING TOOLS DOING SAME THING                  |
|--|------------|-----------|--|
| <b>General Assessment</b>                                    |            |           |  |
| TCP port scanning  | YES        | YES       | nmap, superscan, and numerous others             |
| UDP port scanning  | NO         | YES       | nmap, superscan, and numerous others             |
| Banner grabbing  | YES        | YES       | nmap, superscan, and numerous others             |
| Service enumeration  | NO         | YES       | nmap   |
| Operating system identification                              | NO         | YES       | nmap, xprobe, quaso                              |
| Hostname resolution  | YES        | YES       | nmap, nslookup, numerous others                  |
| ICMP tracerouting  | YES        | YES       | traceroute, visualroute, cheops                  |
| TCP tracerouting   | NO         | YES       | firewalk, LAN MapShot                            |
| Firewall detection   | NO         | YES       | cheops, LAN MapShot                              |
| Router detection   | NO         | YES       | ISS, nessus, LAN MapShot, InFlow, many more      |
| Database detection   | NO         | YES       | AppDetective, ISS                                |
| Wireless device detection                                    | NO         | YES       | ISS, nessus                                      |
| <b>General Defense</b>                                       |            |           |  |
| ISAPI filter   | YES        | NO        | URLScan  |
| Attack signature recognition-based web server defense        | YES        | NO        | Snort  |
| <b>Vulnerability Checking</b>                                |            |           |  |
| Network level checks   | NO         | YES       | nessus, ISS, cybercop, whisker, retina + more    |
| Operating system checks                                      | NO         | YES       | nessus, ISS, cybercop, many more                 |
| Remote windows registry checks                               | NO         | YES       | nessus, ISS, cybercop, many more                 |
| Web server checks  | YES        | YES       | nessus, ISS, cybercop, whisker, retina           |
| [Web] application checks                                     | NO         | YES       | owasp tools, Wharsenal, whisker, retina          |
| Wireless device checks                                       | NO         | YES       | nessus, ISS                                      |
| Vulnerability risk rating                                    | NO         | YES       | nessus, CyberCop, ISS, Shadow, many more         |
| Custom Scripting language for vulnerability checks           | NO         | YES       | nessus (nasl)                                    |
| Interpreter for parsing and executing the scripting language | NO         | YES       | nessus (nasl)                                    |
| <b>Data Storage</b>  |            |           |  |
| Enterprise database usage (MS SQL)                           | NO         | YES       |  |
| Data Encryption  | NO         | YES       |  |
| Segregation of data by customer                              | NO         | YES       |  |
| Segregation of data by user account                          | NO         | YES       |  |
| <b>Online Data Presentation and Management</b>               |            |           |  |
| Web based portal with access control                         | NO         | YES       |  |
| Online account management                                    | NO         | YES       |  |
| Integrated remediation management/workflow system (VulnTrak) | NO         | YES       |  |
| Query based data searching (database)                        | NO         | YES       |  |
| Scan scheduler   | NO         | YES       | Retina, ISS, many more                           |
| Automated scan instantiation (starts scans)                  | NO         | YES       | Retina, ISS, many more                           |
| Automated/configurable web based alerting                    | NO         | YES       |  |
| Automated/configurable email Alerting                        | NO         | YES       | Retina   |
| Multi-tier account creation for granular access control      | NO         | YES       |  |
| Interface for complete system management                     | NO         | YES       | ISS, Retina, CyberCop, Shadow, many more         |
| Secure communication over SSL (encrypted traffic)            | NO         | YES       |  |
| Includes third party security intelligence reports           | NO         | YES       |  |
| <b>Reporting</b>   |            |           |  |
| Summary level reporting                                      | YES        | YES       | ISS, CyberCop, nessus, many other                |
| Detailed reporting on a feature basis                        | NO         | YES       | nessus, ISS                                      |
| Network mapping - host level                                 | YES        | YES       | cheops, CyberCop, LAN MapShot, InFlow, many more |
| Network mapping - firewall level                             | NO         | YES       | cheops, LAN MapShot, InFlow, many more           |
| Network mapping - router level                               | NO         | YES       | cheops, LAN MapShot, InFlow, many more           |
| Network mapping - dual homed devices                         | NO         | YES       | cheops, LAN MapShot, InFlow, many more           |
| Data trending (plot historic data points on graph)           | YES        | YES       |  |
| <b>Miscellaneous</b>   |            |           |  |
| Runs only from the command line (i.e. the "C:\>" prompt)     | YES        | NO        | nessus, nmap                                     |

|   |  |  |                                  |
|---|--|--|----------------------------------|
| Runs as a network service   | NO   | YES  | ISS                              |
| Distributed system- multiple servers geographically based                         | NO   | YES  | ISS                              |
| Able to run numerous simultaneous scans   | NO   | YES  | ISS, CyberCop, many more         |
| Security scoring to indicate risk posture   | NO   | YES  |                                  |
| Available as a managed service  | NO   | YES  | ISS                              |
| Available as software   | YES  | YES  | All                              |
| No installation required, just copy files to desired directory                    | YES  | NO   | nmap                             |
| Requires a trained consultant for configuration and installation                  | NO   | YES  |                                  |
| Requires system management to ensure operation                                    | NO   | YES  |                                  |
| Auto-update feature to auto download system updates                               | NO   | YES  | ISS, CyberCop, Retina, many more |
| Requires third party enterprise software to operate                               | NO   | YES  |                                  |
| <b>Marketing &amp; Sales</b>  |  |  |                                  |
| Target market   | Security and systems administration professionals looking for lightweight tools to perform discrete tasks (such as finding web servers). | Large enterprises that require comprehensive "vulnerability management" solution.                                |                                  |
| Price   | Free, or single user license may be purchased for \$150.   | Published pricing begins at \$30,000. Individual sales generally run \$100,000-700,000 for technology licensing. |                                  |
| *NOTE - This feature list comprises all known FoundScan and Fire & Water features |  |  |                                  |

1 Steven Brower, State Bar #93568  
2 Brian Barrow, State Bar #177906  
3 **STEPHAN, ORINGER RICHMAN & THEODORA**  
4 A Professional Corporation  
5 535 Anton Boulevard, Suite 800  
6 Costa Mesa, California 92626  
7 Telephone: (714) 241-0420  
8 Attorneys for Defendants

8 SUPERIOR COURT OF THE STATE OF CALIFORNIA  
9 FOR THE COUNTY OF ORANGE

10  
11 FOUNDSTONE, INC.,

12 Plaintiff,

13 vs.

14 NT OBJECTIVES, INC., et al.,

15 Defendants.

CASE NO: 02CC15350  
Judge David H. Brickner  
Dept. C17

DECLARATION OF JASSEN D.  
GLASER IN OPPOSITION TO OSC RE:  
PRELIMINARY INJUNCTION

Date: October 25, 2002  
Time: 3:00 p.m.  
Dept: C17

17  
18 I, Jassen D. Glaser, declare:

19 1. I am a defendant in this action. The facts stated herein are of my own personal  
20 knowledge except for those set forth on information and belief, and as to any such facts, I have a  
21 basis for believing that they are true. If called upon to testify as a witness I could and would  
22 competently do so as set forth herein.

23 **ROLE IN THIS LITIGATION**

24 2. I am individually named as a defendant in this matter. I am also the president, director  
25 and majority shareholder of defendant NT Objectives, Inc. ("NTO").

26 3. I was previously employed by Plaintiff Foundstone, Inc. for approximately two years  
27 as more fully outlined below. I was never an officer or director of Foundstone. That is, while my  
28 title was "Director of Engineering" I was never a director in the legal sense of service on the Board

D:\wpdocs\sb102110.02

1

DECLARATION OF JASSEN D GLASER IN OPPOSITION TO OSC RE: PI

1 NTO. I told him that I wanted to develop one of the products which I had previously sold to  
2 Foundstone, VisualLast, which Foundstone was obligated to return to me under oral agreement. I  
3 even asked him to invest \$250,000 in NTO. In fact, I have several emails from Jeanne Miller-  
4 Romero, the inhouse legal counsel at Foundstone, as late as 3 months after I resigned from  
5 Foundstone, saying that she is working on getting an agreement approved to return that software to  
6 me. However, I ultimately realized that Foundstone was not going to proceed with the oral  
7 agreement.

#### 8 FOUNDSTONE'S ALLEGED TRADE SECRETS

9 26. I have read the Declaration of Stuart McClure submitted in support of this OSC re:  
10 Preliminary Injunction. From review of that document I am unable to determine any specific  
11 method, algorithm or technique which is being referred to by Foundstone as a trade secret.

12 27. As stated above, the work which I did while I was employed by Foundstone was good  
13 programming, but the methods, techniques and algorithms which I used were the same as those I  
14 would expect would be used by any other excellent experienced programmer.

15 28. The McClure declaration tries to give the impression that there are real technology  
16 secrets at Foundstone. I would, almost without exception, disagree, as more fully set forth below.

17 29. As differentiated from the technical area, I would agree that Foundstone probably has  
18 legitimate trade secrets regarding certain aspects of their customer relationships, their pricing  
19 strategy, and their financial affairs. However, those items are simply not implicated here. We are  
20 offering a different product and/or service to a different market. Foundstone's minimum service  
21 packages are over \$30,000 and it was my impression that they really were not interested in packages  
22 of less than \$50,000. NTO's "Fire & Water" package is free to individuals and \$125 for  
23 organizations. It is not just a matter of degree. We do not offer a service which replaces the service  
24 offered by Foundstone. While we would be willing to do business with the same companies which  
25 are customers of Foundstone, any resulting business we received would be in addition to, not instead  
26 of, any service which those companies were obtaining from Foundstone.

27 30. A simple example might help to explain the material difference between the business  
28 of Foundstone and the business of NTO. The tools which NTO intends to provide as part of "Fire &

D:\wpdocs\sb102110.02



1 Water" are solely for stand-alone use. In contrast, none of Foundscan can be used without  
2 connecting the software to an Enterprise SQL database (a separate special purpose computer with  
3 special purpose software).

4 31. Foundstone does have real competitors. Those companies would include Qualys,  
5 Guardent and Vigilante. Some of those companies are much larger than Foundstone and claim to  
6 offer many more capabilities than Foundstone. Those companies already claim to have the  
7 technology and the ability to do what is done by Foundstone. To the extent they do not have such  
8 capabilities it is because they have not undertaken the expense or the effort to write the software,  
9 but it is not because there are "secrets" that they don't know or can't obtain from public sources.

10 32. We have, as a result of this litigation, been required to spend time meeting with our  
11 legal counsel. Based on those discussions, and based on my discussions with the other employees  
12 of NTO, and based on my knowledge of computer programming and computer techniques, neither  
13 I, nor any other person at NTO, is utilizing anything which I understand to constitute a trade secret  
14 of Foundstone in the furtherance of the business of NTO.

15 33. For every technology employed by Foundstone I can find and reference a pre-existing  
16 public source for that technology. This would specifically include, but not be limited to, clear  
17 instruction on how to do network topology maps. To the best of my knowledge, during the time I  
18 was employed by Foundstone we didn't invent any new algorithms or technological advancements.

#### 19 DECISION TO LEAVE FOUNDSTONE

20 34. After almost two years at Foundstone I realized that I did not have any real future or  
21 potential for advancement at Foundstone. I decided to resume working for myself at NTO. I  
22 advised Stuart McClure that I would be resigning. My last day at Foundstone was May 3, 2002.  
23 In fact, on that date, I "gave" a brand new software tool to Foundstone as a departing present. I am  
24 informed and believe that the software tool has been utilized by Foundstone to enhance its  
25 marketing efforts and is currently downloadable from their website.

26 35. I am aware that under my employment agreement with Foundstone I am not allowed  
27 to solicit any Foundstone employee to work for NTO. Whether or not such term is legally valid I  
28 have fully complied with the spirit of the requirement. I have never contacted any employee of

D:\wpdocs\sb102110.02

7

DECLARATION OF JASSEN D GLASER IN OPPOSITION TO OSC RE: PI

1 Foundstone to suggest that they work for me and/or NTO. However, our software development  
2 group was relatively small and I was friends with many people at the company. When people who  
3 are employees of Foundstone have contacted me to find out whether they can work for NTO I have  
4 reminded them of the non-solicitation clause and I have told them that I can't make any agreements  
5 with them while they are employees of Foundstone.

6 **ADDITIONAL ITEMS OF INFORMATION**

7 36. The Declaration of Stuart McClure (§ 30-32) refers to a presentation which I made at  
8 the Black Hat conference on July 29, 2002. It is phrased in a strange way because Mr. McClure was  
9 not actually there during my presentation. He is just reporting what he was told by someone else.  
10 Significantly, the presentations, including the one I gave with Michael Morton, are videotaped.  
11 After this litigation started we ordered a videotape of the presentation which our counsel will make  
12 available for review, by the Court, upon request. Within the past week I have reviewed everything  
13 which was said by me, and by Mr. Morton, on that videotape. There is nothing which we said which  
14 reveals or refers to any trade secrets of Foundstone.

15 37. In the interest of clarity, I will address one specific hearsay statement by Mr. McClure.  
16 He says in his Declaration (§ 31) that one of the functions demonstrated at the Black Hat conference,  
17 which "mimicked the proprietary functions" of FoundScan was "determining and graphically  
18 mapping a computer network in a manner nearly identical to FoundScan." Attached hereto as  
19 Exhibit "D" is the graphical mapping of a computer network as performed by FoundScan. This  
20 example is intentionally publicly disclosed by FoundStone on their website. Attached hereto as  
21 Exhibit "E" is the version which is posted on the NTO website and which was shown at the Black  
22 Hat conference which Mr. McClure describes.

23 38. In the Declaration of Stuart McClure (§ 32) he says that it would have been impossible  
24 for us to create the Fire & Water toolkit in three months. I would note the following. First, while  
25 we announced the toolkit at that time, it clearly wasn't finished. It was more than two months later  
26 that we first said we would be ready to make that tool available. Second, while I was at Foundscan  
27 I saw a PowerPoint presentation, done by Mr. McClure, which showed that my development teams  
28 were capable of preparing computer code at impressive rates, not because of any technology at

D:\wpdocs\sb102110.02

8

DECLARATION OF JASSEN D GLASER IN OPPOSITION TO OSC RE: PI

1 Foundscan, but because of my abilities and those of my staff. Third, as set forth in the declaration  
2 of Michael Morton, the various graphical mapping capabilities which exist in the Foundscan  
3 software are almost exclusively the byproduct of Mr. Morton's prior work on his rayscan project  
4 or are otherwise disclosed in the books to which he references. Fourth, had Mr. McClure been  
5 present at the presentation he is describing to the Court he would have known that the graphical  
6 displays which NTO intends to distribute do not mimick the proprietary functions of Foundscan.

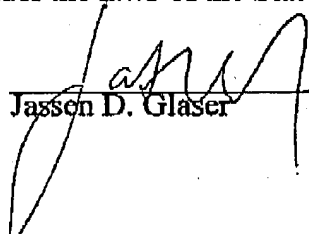
7 39. The Declaration of Stuart McClure makes reference to Operating System  
8 Identification. We have elected not to provide the court with a mountain of paper showing how  
9 much of the technology, generally referred to by Foundstone, is in the public domain. However, in  
10 the interest of clarity, attached hereto as Exhibit "F" is Version 2.5 of the paper "ICMP Usage in  
11 Scanning" by Ofir Arkin, the founder of the Sys-Security Group. Ofir Arkin does not have any  
12 affiliation with Foundstone or with NTO. This paper can be obtained as a free download, without  
13 any registration, payment or agreement to any conditions. This document discloses, to those who  
14 take the time to understand the document, how to do what is referred to by Mr. McClure as OS  
15 Identification. I know that this document forms the basis for the OS Identification in the FoundScan  
16 product because various versions of this document were used by my development team in order to  
17 implement OS Identification at Foundstone.

18 40. Further, attached hereto as Exhibit "G" is the home page from [www.sys-security.com](http://www.sys-security.com),  
19 the organization operated by Ofir Arkin. At the bottom of the page it refers to Xprobe2 "an active  
20 operating system fingerprinting tool with a different approach to operating system fingerprinting.  
21 Xprobe2 rely on fuzzy signature matching, probablistic guesses, multiple matches simultaneously,  
22 and a signature database." It also indicates that you can download the source code, for free, from  
23 this site. We are not arguing that this is the exact same thing as what Foundstone is offering. In  
24 fact, in some ways it may even be more sophisticated than Foundstone. Our concern is that when  
25 Foundstone claims to "own" OS Identification, or any of their other alleged trade secrets, without  
26 defining what they have which is different or special (i.e. - which might qualify as a trade secret),  
27 we can't show that the specific item: a) is not, in fact, a secret; or b) is not, in fact, used by NTO.

28 D:\wpdocs\sb102110.02

1        41. While we reserve the right to use any non-proprietary technology in the future,  
2 including OS Identification, it should be clearly noted that the "Fire & Water" package does not  
3 include any OS identification and has never been intended to include such capability.

4        I declare, under penalty of perjury, that the foregoing is true and correct and that this  
5 declaration is executed on October 23, 2002, under the laws of the State of California.

6  
7   
8 Jassen D. Glaser  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

D:\wpdocs\sb102110.02

10

DECLARATION OF JASSEN D GLASER IN OPPOSITION TO OSC RE: PI

1 Steven Brower (State Bar No. 93568)  
2 Brian P. Barrow (State Bar No. 177906)  
3 STEPHAN, ORINGHER, RICHMAN & THEODORA, P.C.  
4 535 Anton Boulevard, Suite 800  
5 Costa Mesa, California 92626-1902  
6 Telephone (714) 241-0420  
7 Telecopier (714) 241-0622

8 Attorneys for Defendants  
9 NT OBJECTIVES, INC., J.D. GLASSER,  
10 MICHAEL MORTON, ERIK CASO and  
11 DAN KUYKENDALL

12 **SUPERIOR COURT OF THE STATE OF CALIFORNIA**  
13 **FOR THE COUNTY OF ORANGE**

14 FOUNDSTONE, INC., a Delaware  
15 corporation

16 Plaintiff,

17 v.

18 NT OBJECTIVES, INC., a California  
19 corporation; J.D. GLASER, an individual;  
20 MICHAEL MORTON, an individual; ERIC  
21 CASO, an individual; DAN  
22 KUYKENDALL, an individual; and DOES  
23 1 through 50, Inclusive.

24 Defendants.

Case No. 02CC15350

ASSIGNED FOR ALL PURPOSES TO:  
HONORABLE DAVID H. BRICKNER  
DEPT: C17

MEMORANDUM OF POINTS AND  
AUTHORITIES IN SUPPORT OF  
DEFENDANTS' OPPOSITION TO  
PLAINTIFF'S OSC RE:  
PRELIMINARY INJUNCTION

HEARING DATE: October 25, 2002  
TIME: 3:00 p.m.

DEPT: C17  
TRIAL DATE: None

COMPLAINT FILED: October 2, 2002

25 **MEMORANDUM OF POINTS AND AUTHORITIES**

26 **I. INTRODUCTION**

27 Plaintiff Foundstone, Inc. seeks to enjoin its former employees from releasing "Fire &  
28 Water Toolkit," a computer software product that Foundstone has never seen, but which  
allegedly incorporates misappropriated trade secrets. Foundstone fails to identify the supposedly  
stolen trade secrets or specify how they are supposedly being used in defendants' product,  
instead relying on an allegation that defendants could only have developed their product by  
virtue of their prior employment at Foundstone.

1 allegation its "algorithms, methods, and databases" are trade secrets thus does not describe  
2 which, if any, of the components of "Fire and Water Toolkit" were supposedly misappropriated  
3 from Foundstone. Without such descriptive information, Foundstone fails to demonstrate any  
4 likelihood of success on the merits or make any showing of relative interim harm.

5 **3. Foundstone Also Refuses To Provide Further Specificity Of Its Alleged**  
6 **Trade Secrets As Required By the Discovery Act.**

7 Code of Civil Procedure section 2019, subdivision (d), states the following with regard  
8 to identification of trade secrets prior commencing discovery:

9 In any action alleging the misappropriation of a trade secret . . . , before  
10 commencing discovery relating to the trade secret, the party alleging the  
11 misappropriation shall identify the trade secret with reasonable particularity  
12 subject to any orders that may be appropriate under [the Uniform Trade Secret  
13 Act].

14 Defendants have repeatedly asked Foundstone for such an identification, specifically  
15 explaining that Foundstone's pleadings contain an insufficient description of the alleged trade  
16 secrets. Tellingly, Foundstone has so far refused to provide defendants with the requested  
17 specificity, despite previously seeking leave to commence the discovery process.

18 **D. Foundstone's "Software, Methods, and Algorithms" Are Not Protectible**  
19 **Trade Secrets.**

20 Even if the generic "software, methods, and algorithms" is considered a sufficient  
21 description of a trade secret, Foundstone nevertheless fails to provide evidence demonstrating  
22 that such material is entitled to trade secret protection. Foundstone thus fails to show that its  
23 claimed "software, methods, and algorithms" are actually trade secrets.

24 "The test for trade secrets is whether the matter sought to be protected is information (1)  
25 which is valuable because it is unknown to others and (2) which the owner has attempted to keep  
26 secret." (*Schlage Lock Company v. Whyte* (2002) 101 Cal.App.4th 1443, 1453, citing *Abba*  
27 *Rubber Co. v. Seaquist* (1991) 235 Cal.App.3d 1, 18.) Foundstone offers no evidence satisfying  
28 either of these requirements.

1                   **1. Foundstone's Alleged Trade Secrets and Software Products Are Based**  
2                   **on Information Available In the Public Domain.**

3           The United States Supreme Court has stated that information that is public knowledge or  
4 that is generally known in an industry cannot be a trade secret. (*Ruckelshaus v. Monsanto Co.*  
5 (1984) 467 U.S. 986, 1002.)

6           As shown in the Declarations submitted in Opposition, Foundstone did not actually invent  
7 anything. To the extent that defendants are able to identify alleged trade secrets the Declarations  
8 show that such materials were originally created by sources outside of Foundstone, are known  
9 to those who are experienced in the industry, or have been deliberately disclosed to the public  
10 by Foundstone.

11                   **2. Foundstone Disclosed Any Trade Secrets Contained In Its Software**  
12                   **When It Provided Unprotected Marketing Versions to the Press.**

13           Foundstone cannot validly claim that its various databases upon which FoundScan  
14 operates have been the subject of efforts to prevent the loss of secrecy. As explained in the  
15 declaration of defendant Erik Caso, Foundstone repeatedly provided members of the press with  
16 copies of its software products that were vulnerable to disclosure. Foundstone never insisted that  
17 the members of the press to whom the software was delivered execute nondisclosure agreements  
18 or otherwise promise not to publicize its alleged trade secrets. As such, members of the press  
19 who received the software were free to, even encouraged to, view the very components of the  
20 software (i.e. databases, etc) that Foundstone now contends are trade secrets they have attempted  
21 to keep confidential. Foundstone's own acts of disclosing information to outsiders demonstrates  
22 that the alleged trade secrets no longer enjoy the secrecy that Foundstone claims in this lawsuit.

23                   **E. Foundstone Has No Evidence of any Actual or Threatened Misappropriation**  
24                   **of "Software, Methods, and Algorithms."**

25           This court may only enjoin "actual or threatened misappropriation" of a trade secret.  
26 (Civ. Code, § 3426.1, subd. (a).) "Misappropriation" is defined in this context as improper  
27 acquisition of a trade secret or its nonconsensual use or disclosure. (Civ. Code, § 3426.1, subd.  
28 (b).) Foundstone is therefore burdened with making a showing that defendants are either

Kuykendall.org :: A place for Kuykendalls to converge

Page 1 of 4

[Create an account](#)[Home](#) • [Topics](#) • [Downloads](#) • [Your Account](#) • [Submit News](#) • [Top List](#)

October 25, 2002

## ★ Main Menu

- [Downloads](#)
- [FAQ](#)
- [Journal](#)
- [Members List](#)
- [Messages](#)
- [News](#)
- [Photo Gallery](#)
- [Polls](#)
- [Recommend Us](#)
- [Reviews](#)
- [Search](#)
- [Stats](#)
- [Submit News](#)
- [Topics](#)
- [Top List](#)
- [Web Links](#)

## ★ Who's Online

We have 20 guests and 1 member online

You are an anonymous user. You can register for free by clicking [here](#)

## ★ Administration

[Admin Page](#)

## ★ Languages

Select interface language:

[English](#)

Welcome to the Kuykendall Website.

Hello, I am Dan Kuykendall ( aka Seek3r ). Welcome to my website. This site is intended to be for the use of the visitors and myself to discuss whatever issues are on our minds.

## ★ Legal Docs: The response to their complaint



In this article on Fool.com there is a great example of what Foundstone is doing:

'Finally, there's bluffing. This happens all the time. During the 1992 presidential campaign, Ross Perot copyrighted his likeness and convinced Dana Carvey to stop parodying him on Saturday Night Live, despite the fact that parody explicitly falls under "fair use" and they had been parodying other copyrighted and trademarked properties (such as Eddie Murphy doing "Gumby" and "Buckwheat") for years. The important thing wasn't whether Perot had a case that would hold up in court, it's that he had a lot of money to spend on lawyers, he sounded serious, and that Dana Carvey backed down.'

The fact is that we have done nothing wrong, but they have money and act serious to intimidate us. The problem for them is... We are not intimidated!

Here is the full collection of legal docs:

**The new stuff:**

Here are the declarations of all the defendants in the case; JD Glaser, Mike Morton and myself. Here is our [Defendants Memorandum of Opposition](#) and their final reply. Now its in the judges hands and his decision will be posted on [his website](#) by 4:30PM PST, Oct 25 2002.

**The old stuff:**

The Temporary Restraining order and Stuart McClure's Declaration. Their complaint - 1-10 / 11-20 / 21-23 and the Plaintiffs Memorandum - 1-10 / 11-16

Click the read more link for **\*my\*** opinions on each point of Stuart's Declaration.

Posted by: seek3r on Thursday, October 24, 2002 - 11:46 AM PST  
[Read more...](#) (9283 bytes more)

## ★ Legal Case: Here comes the press



Things are moving along with the lawsuit. Friday is going to be the day that the Judge will make the decision on the Preliminary Injunction. If we win, then we will be free to release our toolkit. The press has gotten ahold of the story as well. It has shown up on the very popular [Info Security Wire](#) and on [Security Administrator](#).

Click the [Read more...](#) link for some internal problems at Foundstone, that stem from this case.

Posted by: seek3r on Tuesday, October 22, 2002 - 12:07 AM PST  
[Read more...](#) (681 bytes more)

## ★ Job at

- [NT OBI](#)
- [phpGro](#)
- [Clear R](#)

## ★ Surve

Should

- Yes, will
- I dunnc
- No, its i
- choice
- Is that i
- desert pl

## [ Res

Co

## ★ User's

Usernam

Passwor

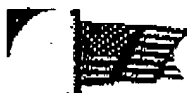
☐ Rem

Don't ha  
yet? You  
As a regl  
have son  
like a the  
comment  
and post  
with your

EXHIBIT A PAGE 1 OF 4



Legal Docs: The response to their complaint :: Kuykendall.org :: A place for Kuykendalls... Page 1 of 4

[Create an account](#)[Home](#) • [Topics](#) • [Downloads](#) • [Your Account](#) • [Submit News](#) • [Top List](#)

October 25, 2002

## ★ Main Menu

- [Downloads](#)
- [FAQ](#)
- [Journal](#)
- [Members List](#)
- [Messages](#)
- [News](#)
- [Photo Gallery](#)
- [Polls](#)
- [Recommend Us](#)
- [Reviews](#)
- [Search](#)
- [Stats](#)
- [Submit News](#)
- [Topics](#)
- [Top List](#)
- [Web Links](#)

## ★ Who's Online

We have 20 guests and 0 members online

You are an anonymous user. You can register for free by clicking [here](#)

## ★ Administration

- [Admin Page](#)

## ★ Languages

Select interface language:

[English](#)

Legal Docs: The response to their complaint  
Posted by: seek3r

In this article on [Fool.com](#) there is a great example of what Foundstone is doing:  
'Finally, there's bluffing. This happens all the time. During the 1992 presidential campaign, Ross Perot copyrighted his likeness and convinced Dana Carvey to stop parodying him on Saturday Night Live, despite the fact that parody explicitly falls

under "fair use" and they had been parodying other copyrighted and trademarked properties (such as Eddie Murphy doing "Gumby" and "Buckwheat") for years. The important thing wasn't whether Perot had a case that would hold up in court, it's that he had a lot of money to spend on lawyers, he sounded serious, and that Dana Carvey backed down.'

The fact is that we have done nothing wrong, but they have money and act serious to intimidate us. The problem for them is... We are not intimidated!

Here is the full collection of legal docs:

**The new stuff:**

Here are the declarations of all the defendants in the case; JD Glaser, Mike Morton and myself. Here is our Defendants Memorandum of Opposition and their final reply. Now its in the judges hands and his decision will be posted on [his website](#) by 4:30PM PST, Oct 25 2002.

**The old stuff:**

The Temporary Restraining order and Stuart McClure's Declaration.

Their complaint - 1-10 / 11-20 / 21-23 and the Plaintiffs Memorandum - 1-10 / 11-16

Click the read more link for \*my\* opinions on each point of Stuart's Declaration.

My quick response to [Stuart McClure's Declaration](#) which is the only actual "facts" they have presented to the court.

\*These are my opinions and while I believe everything I am stating to be true, this is not a legal rebuttal.\* I will be posting all our legal responses as they become available.

First lets look at what the law considers a Trade Secret:

A good explanation basidy says that "A trade secret is any information that allows you to make money because it is not generally known". Since hacker techniques are generally known and published on several websites, as well as techniques for port scanning, operating system fingerprinting and visualized network mapping from traceroute data, they cannot claim them as Trade Secrets.

Read my responses along side of his Declaration.

- 1) True enough
- 2) Notice how he makes it specific who their target audience/market is. This is not the same as NIO's which is releasing small command line tools for the general network and security admin.
- 3) No problem with this

## ★ Related links

- [More about Legal Issues](#)
- [News by seek3r](#)

Most read story in Legal Issues:  
[My Right to work](#)

EXHIBIT A PAGE 2 OF 4

Legal Docs: The response to the complaint :: Kuykendall.org :: A place for Kuykendalls... Page 2 of 4

4) I question the actual costs/hours. From my calculations a single person's full time hours for a year come out to about 2,000. To account for 80,000 would mean 20 people (\$100k average annual salary/cost) working on it full time for the last two years its been in development.

Doesnt seem accurate... but its not an important point to the case... but one that he repeats several times, so I felt I should shoot it down up front.

5) For sure their source code, and databases are their own. Possibly the actual algorithm for calculating their often discussed "FoundScore". Other than that I am not personally aware of any other Trade Secrets that they have.

6) He says in this that FoundScan is the heart of Foundstone's business. This does not make sense in terms of revenue as far as I know. Foundstone has been a very successful "Consulting and Training" company up until their recent release of FoundScan. I dont see how a new product that has only sold a limited number of copies has outstripped their long established consulting and training business.

7) What he is talking about is operating system "fingerprinting" for "OS Ident". This is a very common and well understood process in the industry. Yes they have their proprietary implementation, but "fingerprinting" and "OS Ident" (which we dont even do yet) is certainly not some Trade Secret to Foundstone.

8) Again, they have their own proprietary implementation, but visually mapping a network based on traceroute data is well understood and exists in many products. In Stu's own book he talks about Cheops which does this and has existed far longer than FoundScan.

9) Again, they have their own proprietary implementation and technique for automated vulnerability testing, but as far as I know all of the vulnerability checks they (and just about anyone else) do are well described on sites such as Security Focus. They then display this in a spherical map which they have examples of on their own website... how he still thinks this is a trade secret is a mystery to me. Regardless, this is not something we are even doing.

10) Again, they have their own proprietary implementation, but this is just talking about a web interface to an application. If I had stolen their code, this is a problem. Finally, I am not even doing anything even remotely similar for NTO.

11) Again, they have their own proprietary implementation, but web crawling to inventory and hack is a well understood and documented process in the security industry.

12) Again, they have their own proprietary implementation, but most of this is just trend reporting. Stuff that tools like Microsoft Excel and Crystal Reports have been doing for years. Trending is trending. The mentioned "scoring system" \*is potentially\* a secret algorithm, but one that they talk about the details of in the Hacking Exposed book and other forums. But I do think the exact algorithm itself is a secret... one which I dont even know. This is also not something we are doing at NTO.

13 - 17) I have no problem with these

18) This is not entirely true. I wont go into the details... but up till very recently some parts were plain text. Regardless, the general point is true enough.

19 & 20) No problem with these

21) This is not true. The documents I have seen from the purchase do not cover NTO Scanner, so the word "all" is not accurate. Whats more is that scanning is the heart of the Fire & Water toolkit. So we already had code to work from, I dont know if any of the old scanner code was actually used for the new scanner, but we had some to use none the less.

22 & 23) No problem with these

24) This is kind of lame and as far as Im aware they knew

EXHIBIT A PAGE 3 OF 4

Legal Docs: The response to their complaint :: Kuykendall.org :: A place for Kuykendalls... Page 3 of 4

about his desire to bring NTO back to life. Additionally NTO is \*not\* a direct competitor to Foundstone. We happen to be another computer security tool company... but that does not make us a direct competitor since we are not targetting their Fortune 500 market with a like product.

25 - 30) No problem with these

31) He calls the 3 functions, "proprietary functions" which means that the given functionality and not just their implementation is proprietary. I am pretty sure the only way to make a "function" proprietary is to get a Patent, which they do not have and could not get because of the vast amount of prior art. Point 1 is just a network map from traceroute data, as mentioned before. Also he calls our map nearly identical to their map. See for yourself his exaggeration. Number 2 seems to indicate that using a hypertext solution for reporting (aka HTML) is not allowed. For one, we dont even have such a tool,, for another we almost exclusively use XML and XSLT for all of our data and reporting, AND its not a trade secret to use HTML for a report! Number 3 is again not something we even have a product to do, and its no trade secret.

32) I take great personal issue with this. The NTO teams happens to be some of the best from their own development team. I have proven to him and the entire development team on a couple occasions what I can accomplish in a short period of time. I wrote their entire web interface to scan management in about 3 weeks when I first started with the company. I also wrote their vulntrak app in about a month. The other developers at NTO are even probably even better programmers than I am. This is really just a personal insult... and its also not true. We have produced rapidly... but even then we had just barely been ready to release the products. What we showed at Blackhat was demo's and mock-ups for the most part.

33) No problem with this

34) This is so lame. First the security industry has been built on disclosure of techniques for hacking and securing. The point that their Fortune 500 customers would forgo buying a \$150k software package that integrates over a hundred features and provides enterprise scalability and management, for a set of DOS PROMPT/Command Line utilities is a joke!

35) Yes, so we are not rich :-)) I do have a problem with that... but thats not an issue for this :-))

36) I have made my case against this lame point, but I will wait for him to explain Foundstone's Free tools and Stuarts own book called Hacking Exposed which explains in great detail the methods of hacking.

37 - 39) No problem with these.

In the end I hope I have explained how weak their case is. I will state again that "WE HAVE NOT STOLEN THEIR CODE OR ANYTHING THAT IS THEIR ACTUAL TRADE SECRETS". We have our own ideas, our own code and our own products. I wish Foundstone would just leave us alone, stop bluffing and trying to make us burn our money on the legal costs for defending ourselves. I encourage anyone that wants to, to email them and give them your opinions (pro or con, that is your choice).

I do hope you encourage them to back off and let us go about our business and them go back to theirs.

Dan Kuykendall

Legal Docs: The response to their complaint | Login/Create an account | 0 Comments

Threshold ☐ Thread ☒ Oldest first ☐ Newest first ☐

Comments are owned by the poster. We aren't responsible for their content.

[Home](#) • [Topics](#) • [Downloads](#) • [Your Account](#) • [Submit News](#) • [Top List](#)

EXHIBIT A PAGE 4 OF 4

**FILED**  
 SUPERIOR COURT OF CALIFORNIA  
 COUNTY OF ORANGE  
 CENTRAL JUSTICE CENTER

OCT 04 2002

ALAN SLATER, CLERK OF THE COURT  
*Young*  
 BY J. YOUNG

Darrell L. Olson, Bar No. 77,633  
 Michael K. Friedland, Bar No. 157,217  
 Douglas T. Hudson, Bar No. 210,385  
 KNOBBE, MARTENS, OLSON & BEAR, LLP  
 2040 Main Street  
 Fourteenth Floor  
 Irvine, California 92614  
 (949) 760-0404 (telephone)  
 (949) 760-9502 (facsimile)

Attorneys for Plaintiff FOUNDSTONE, INC.

SUPERIOR COURT OF THE STATE OF CALIFORNIA  
 COUNTY OF ORANGE, CENTRAL JUSTICE CENTER

FOUNDSTONE, INC., a Delaware  
 corporation,

Plaintiff,

v.

NT Objectives, Inc., a  
 California corporation; J.D.  
 Glaser, an individual; Michael  
 Morton, an individual; Eric  
 Caso, an individual; Dan  
 Kuykendall, an individual; and  
 DOES 1 through 50, inclusive,

Defendants.

CASE NO. 02CC1530

ASSIGNED FOR ALL PURPOSES TO:  
 COMMISSIONER ELEANOR M. PALK  
 DEPT. C62

DECLARATION OF STUART MCCLURE  
 IN SUPPORT OF PLAINTIFF'S  
 APPLICATION FOR TEMPORARY  
 RESTRAINING ORDER AND  
 PRELIMINARY INJUNCTION,  
 REQUEST FOR EXPEDITED  
 DISCOVERY, AND REQUEST FOR  
 FILING UNDER SEAL.

HEARING DATE: Oct. 4, 2002

TIME: 1:30 p.m.

DEPT: C62

TRIAL DATE: None.

COMPLAINT FILED: Oct. 2, 2002

I, Stuart McClure, declare:

1. I am the President and Chief Technology Officer of  
 Foundstone, Inc., Plaintiff in the above-titled action. I  
 have served as President and Chief Technology Officer of

1 Foundstone since 1999. The following is true of my own  
2 personal knowledge; and if called and sworn as a witness, I  
3 could and would testify competently thereto.

4 2. Foundstone specializes in computer network security  
5 and provides network security audits, education services, and  
6 security software for Fortune 500 companies, government  
7 agencies, and others worldwide.

8 3. Foundstone's principal product is FoundScan, a  
9 unique proprietary software system that automatically scans  
10 computer networks, tests networks for "hacking"  
11 vulnerabilities, provides detailed graphical reports and maps  
12 of tested networks and vulnerabilities, scores the security of  
13 networks, and reports on results of those tests.

14 4. Over the past three years, Foundstone has invested  
15 more than \$4 million dollars and more than 80,000 person-hours  
16 in research, development, and testing of the proprietary  
17 FoundScan software.

18 5. Foundstone's proprietary FoundScan software contains  
19 a number of well-guarded algorithms, methods and databases  
20 that are highly valuable trade secrets of Foundstone. In  
21 particular, FoundScan contains trade secret software  
22 technology relating to securing corporate and government  
23 computer networks against hackers, cyber-terrorists, and  
24 electronic espionage.

25 6. This trade secret technology is the heart of  
26 Foundstone's business, and its value would be substantially  
27 diminished if it were to fall into the public domain. So long  
28 as the technology continues to be maintained as secret,

1 competitors and potential competitors would have to invest  
2 similarly large amounts of time and money to develop systems  
3 with similar capabilities to compete.

4 7. Foundstone's proprietary methods and databases for  
5 operating system identification include techniques for sending  
6 TCP (Transmission Control Protocol) and ICMP (Internet Control  
7 Messaging Protocol) packets to a target computer, which then  
8 responds with packets of its own. Based on the target  
9 computer's response, the operating system of the target  
10 computer is identified against a proprietary database.

11 8. Foundstone's proprietary methods and databases for  
12 determining and graphically mapping a computer network include  
13 methods for ICMP and TCP tracerouting to all devices,  
14 analyzing the results of the tracerouting to understand the  
15 placement of those devices on the network, and then visually  
16 displaying those devices in a three-dimensional map.

17 9. Foundstone's proprietary methods and databases for  
18 detecting network vulnerabilities and reporting them with the  
19 network map include methods for sending a series of packets to  
20 a target device to prompt a particular response. Once the  
21 response is collected, it is analyzed and compared to a series  
22 of rules that define a vulnerable device. Once identified as  
23 vulnerable, the device is then associated with a device  
24 discovered during the network mapping stage, and the result is  
25 displayed on the three-dimensional spherical map.

26 10. Foundstone's proprietary methods and databases for a  
27 remote administration web interface include methods for  
28 alerting a user of newly discovered vulnerabilities,

proprietary databases of vulnerabilities, tracking of discovered vulnerabilities through a workflow process of fixing them, displaying reports of found vulnerabilities, displaying threat information, displaying and controlling the status of scans, managing user and administrative roles within the web interface, and searching the proprietary database for relevant vulnerability information.

11. Foundstone's proprietary web modules for vulnerability testing of web servers include methods for "crawling" a web site for links, inventorying the technologies in place, "brute forcing" authentication mechanisms to unearth easy-to-guess passwords, guessing the names of existing but not readily linked-to files, testing for poor script input validation, and testing for source code disclosure issues.

12. Foundstone's proprietary methods and databases for reporting vulnerability results over time include methods for an objective security scoring mechanism, a breakdown of network inventory by live hosts, services open, a vulnerability network map, operating systems running, vulnerabilities found, web module analysis, and trending of these results over time.

13. Foundstone has taken extensive steps to protect the trade secrets contained in the FoundScan software, as is necessary to protect the substantial economic and competitive value of Foundstone's trade secrets. These secrets provide efficient network scanning, effective graphical mapping of networks and vulnerabilities on those networks, and understandable reports on network security.

1 14. Foundstone protects its software internally with  
2 passwords, "software source safes," firewalls and other  
3 network security measures.

4 15. Foundstone's physical facilities are protected by  
5 security measures including closed-circuit cameras, security  
6 badges, and biometric devices.

7 16. All of Foundstone's customers are under strict non-  
8 disclosure agreements and covenants not to reverse engineer  
9 FoundScan. Attached as Exhibit "A" to this declaration are  
10 true and correct copies of three example non-disclosure  
11 agreements of the type signed by Foundstone's customers.

12 17. FoundScan service customers do not have any access  
13 to the FoundScan software. These customers only interact with  
14 the software through a web interface, while the software runs  
15 from computers controlled by Foundstone.

16 18. The few FoundScan customers who do have access to  
17 the FoundScan software only have access to object code.

18 19. All of Foundstone's employees, including Defendants  
19 J.D. Glaser ("Glaser"), Michael Morton ("Morton"), Eric Caso  
20 ("Caso") and Dan Kuykendall ("Kuykendall"), signed agreements  
21 recognizing the trade secret status of its software and  
22 research. Attached as Exhibit "B" to this declaration are  
23 true and correct copies of the agreements signed by each of  
24 the above-named individuals.

25 20. All of Foundstone's employees, including Defendants  
26 Glaser, Morton, Caso, and Kuykendall, signed non-disclosure  
27 agreements as part of their agreements promising not to  
28 divulge Foundstone's trade secrets, and they stipulated and



1 secrecy." Vacco Industries, Inc. v. Van Den Berg, 5 Cal.App.  
2 4th 34, 50 [quoting Cal. Civ. Code § 3426.1(d)].

3 Misappropriation of a trade secret includes "[d]isclosure  
4 or use of a trade secret of another without express or implied  
5 consent by a person who . . . (B) At the time of disclosure or  
6 use, knew or had reason to know that his or her knowledge of  
7 the trade secret was: . . . (ii) Acquired under circumstances  
8 giving rise to a duty to maintain its secrecy or limit its  
9 use; or by improper means of disclosure including breach of  
10 duty to maintain secrecy." PMC, Inc. v. Kadisha, (2000) 78  
11 Cal.App.4th 1368, 1382-83 (quoting Cal. Civ. Code §  
12 3426.1(b)). Because each of these elements are met here,  
13 Foundstone is likely to prevail on the merits.

14 1. Foundstone Has Established the Existence of Trade  
15 Secrets

16 The Uniform Trade Secrets Act defines a trade secret as  
17 "information, including a formula, pattern, compilation,  
18 program, device, method, or technique, or process that []  
19 derives independent economic value, actual or potential, from  
20 not being generally known to the public or to other persons  
21 who can obtain economic value from its disclosure or use."  
22 Cal. Civ. Code § 3426.1(d). Schlage Lock Co. v. Whyte, 2002  
23 Cal. App. LEXIS 4634, \*13 (Cal. App. 4th Dist. Sept. 12, 2002)  
24 (affirming trade secret status of misappropriated materials).

25 Foundstone's technologies certainly meet this definition.  
26 The Foundstone technologies at issue are:

27 (1) Foundstone's proprietary methods and databases for  
28 operating system identification, which include techniques for

1 sending TCP (Transmission Control Protocol) and ICMP (Internet  
2 Control Messaging Protocol) packets to a target computer,  
3 which then responds with packets of its own. Based on the  
4 target computer's response, the operating system of the target  
5 computer is identified against a proprietary database of over  
6 eight hundred operating system "fingerprints." (McClure  
7 Decl., ¶7.)

8 (2) Foundstone's proprietary methods and databases for  
9 determining and graphically mapping a computer network, which  
10 include ICMP and TCP tracerouting to all devices, analyzing  
11 the results of the tracerouting to understand the placement of  
12 those devices on the network and then visually displaying  
13 those devices in a three-dimensional map. (Id., ¶8.)

14 (3) Foundstone's proprietary methods and databases for  
15 detecting network vulnerabilities and reporting them with the  
16 network map, which include sending a series of packets to a  
17 target device to prompt a particular response. Once the  
18 response is collected, it is analyzed and compared to a series  
19 of rules which define a vulnerable device. Once identified as  
20 vulnerable, the device is then associated with a device  
21 discovered during the network mapping stage, and the result is  
22 displayed on the three-dimensional spherical map. (Id., ¶9.)

23 (4) Foundstone's proprietary methods and databases for a  
24 remote administration web interface, which include alerting a  
25 user of newly-discovered vulnerabilities, proprietary  
26 databases of vulnerabilities, tracking of discovered  
27 vulnerabilities through a workflow process of fixing them,  
28 displaying reports of found vulnerabilities, displaying threat

1 information, displaying and controlling the status of scans,  
2 managing user and administrative roles within the web  
3 interface, and searching the proprietary database for relevant  
4 vulnerability information. (Id., ¶10.)

5 (5) Foundstone's proprietary web modules for  
6 vulnerability testing of web servers, which include "crawling"  
7 a web site for links, inventorying the technologies in place,  
8 "brute forcing" authentication mechanisms to unearth easy-to-  
9 guess passwords, guessing the names of existing but not  
10 readily linked-to files, testing for poor script input  
11 validation, and testing for source code disclosure. (Id.,  
12 ¶11.)

13 (6) Foundstone's proprietary methods and databases for  
14 reporting vulnerability results over time including an  
15 objective security scoring mechanism, a breakdown of network  
16 inventory by live hosts, services open, a vulnerability  
17 network map, operating systems running, vulnerabilities found,  
18 web module analysis, and trending of these results over time.  
19 (Id., ¶12.)

20 These technologies are not generally known to the public  
21 or to those who could obtain economic value from their  
22 disclosure or use. As discussed above, Foundstone created  
23 these technologies only through investing 80,000 person-hours  
24 and \$4,000,000 of research and development. (Id., ¶4.)

25 These technologies also derive value from their secrecy.  
26 The resulting technologies are valuable to Foundstone only  
27 because - and only for so long as - the technologies are  
28 secret. (Id., ¶6.) Foundstone is able to market its software

1 and services effectively because it offers a product that is  
2 unique. (Id., ¶13.) So long as the technologies continue to  
3 be maintained as secrets, competitors and potential  
4 competitors would have to invest similar large amounts of time  
5 and money to develop systems with similar capabilities to  
6 compete with Foundstone. (Id., ¶16.) If the technologies were  
7 to be disclosed, entities could compete with Foundstone  
8 without having to make these investments, thereby devastating  
9 the market position Foundstone established as a result of its  
10 huge research and development investment. (Id.)

11 2. Foundstone Has Taken Far-Reaching Steps to Protect  
12 the Secrecy of its Technologies

13 As described above, Foundstone has taken extensive steps  
14 to protect the confidential status of its trade secrets.

15 Foundstone protects its software internally with  
16 passwords, firewalls, "software source safes," and other  
17 network security measures. (Id., ¶¶13-14.) Foundstone's  
18 physical facilities are protected by security measures  
19 including "biometric" security devices. (Id., ¶15)

20 All of Foundstone's customers are under strict non-  
21 disclosure agreements and covenants not to reverse engineer.  
22 (Id., ¶16 & Ex. A) Foundstone's service customers do not have  
23 any access to the Foundstone's software. These customers only  
24 interact with the software through a web interface while the  
25 software runs on computers controlled by Foundstone. (Id.,  
26 ¶16.) The few Foundstone customers who have access to  
27 Foundstone's software only see object code. (Id., ¶17).

28 All of Foundstone's employees, including the individual

1 Defendants, signed agreements recognizing the trade secret  
2 status of Foundstone's software and research. (Id., ¶19-20 &  
3 Ex. B.) In addition, all of Foundstone's employees, including  
4 the individual Defendants, signed non-disclosure agreements  
5 promising not to divulge Foundstone's trade secrets. (Id.)

6 3. Defendants Indisputably Had Access to Foundstone's  
7 Trade Secrets

8 There can be no dispute that Defendants had access to  
9 Foundstone's trade secrets. Each of the individual Defendants  
10 was employed by Foundstone and, while at Foundstone, had key  
11 roles in developing the trade secret technologies that are at  
12 issue in this Motion. (Id., ¶29.) Defendant NTO was founded  
13 by and consists of the individual Defendants. (Id.)

14 4. Defendants' Release of Their Software Would Breach  
15 Their Duties to Maintain Secrecy

16 Each of the individual Defendants executed employment  
17 agreements that included comprehensive nondisclosure  
18 provisions. (Id., ¶19 & Ex. B.) By releasing software  
19 containing Foundstone's trade secret technologies, Defendants  
20 would certainly be breaching their obligations of  
21 confidentiality pursuant to their agreements with Foundstone.

22 Because the evidence establishes both that the balance of  
23 the hardships weighs dramatically in favor of Foundstone and  
24 that Foundstone is likely to prevail on the merits, this Court  
25 should temporarily restrain and preliminarily enjoin  
26 Defendants from using or disclosing Foundstone's trade  
27 secrets. Vacco Industries, Inc., 5 Cal. App. 4th 34, 53-54;  
28 Merrill Lynch, 2001 U.S. Dist. LEXIS at \*14-\*17.

DESIGNATED LITIGATION ATTORNEY'S COPY  
ORIGINAL IN DOCKETING SYSTEM

Darrell L. Olson, Bar No. 77,633  
Michael K. Friedland, Bar No. 157,217  
Douglas T. Hudson, Bar No. 210,385  
KNOBBE, MARTENS, OLSON & BEAR, LLP  
2040 Main Street  
Fourteenth Floor  
Irvine, California 92614  
(949) 760-0404 (telephone)  
(949) 760-9502 (facsimile)

Attorneys for Plaintiff FOUNDSTONE, INC.

SUPERIOR COURT OF THE STATE OF CALIFORNIA  
COUNTY OF ORANGE, CENTRAL JUSTICE CENTER

FOUNDSTONE, INC., a Delaware  
corporation,

Plaintiff,

v.

NT Objectives, Inc., a  
California corporation; J.D.  
Glaser, an individual; Michael  
Morton, an individual; Eric  
Caso, an individual; Dan  
Kuykendall, an individual; and  
DOES 1 through 50, inclusive,

Defendants.

) CASE NO. 02CC1530

) ASSIGNED FOR ALL PURPOSES TO:  
) HON. DAVID H. BRICKNER  
) DEPT. C17

) PLAINTIFF'S REPLY IN SUPPORT  
) OF MOTION FOR PRELIMINARY  
) INJUNCTION

) HEARING DATE: Oct. 25, 2002  
) TIME: 3:00 p.m.  
) DEPT: C17  
) TRIAL DATE: None

) COMPLAINT FILED: Oct. 2, 2002

PEA IN SUPPORT OF FOUNDSTONE'S MOTION FOR PI

1 grant the preliminary injunction." ):

2 C. The Facts Establish Likelihood Of Success

3 The second prong of the two-part test for injunctive  
4 relief is likelihood of success on the merits. Foundstone has  
5 clearly established that it is likely to succeed.

6 1. Foundstone's Specified Methods and Techniques Are  
7 Protectable Trade Secrets

8 In its moving papers, Foundstone submitted detailed  
9 evidence that its methods and techniques in six specific areas  
10 constituted protectable trade secrets. (Memo. at 9-11.) In  
11 response, Defendants have raised a number of arguments. All  
12 lack merit.

13 (a) The Trade Secrets Are Adequately Defined

14 Defendants argue that Foundstone has not identified its  
15 trade secrets with sufficient particularity. (Opp. at 6-8.)  
16 To make this argument, Defendants ignore the specific  
17 identification of trade secrets contained in Foundstone's  
18 moving papers and claim that Foundstone only generically  
19 stated "that Foundstone's software contains 'algorithms,  
20 methods, and databases.'" (Id. at 6.) This is simply  
21 incorrect. Foundstone's moving papers stated that its trade  
22 secrets related to six specifically described technologies.  
23 The detailed description of these technologies spanned two  
24 full pages of Foundstone's brief. (Memo. at 9-11; McClure  
25 Decl. ¶¶ 5-12.)<sup>1</sup>

26  
27  
28 <sup>1</sup> Defendants' reliance on Diodes, Inc. v. Franzen, 260  
Cal.App.2d 244 (1968) is misplaced. In Diodes, the plaintiff  
simply referred to a "secret process." In contrast,

1 Defendants similarly assert that they are unable to  
2 understand the described trade secrets. (See, e.g., Glaser  
3 Decl. ¶ 26.) Such assertions are implausible, given that the  
4 Defendants themselves allege that they had critical roles in  
5 developing Foundstone's software. (See, e.g., id. ¶ 38.) The  
6 assertions are also thoroughly repudiated by Defendants' own  
7 declaration; The declarations discuss specific methods and  
8 algorithms allegedly implicated by the claimed trade secrets  
9 in great detail. (Caso Decl. ¶¶ 12-18; Glaser Decl. ¶¶ 16-19;  
10 Morton Decl. ¶¶ 16-19.)<sup>2</sup> See Whyte v. Schlage Lock Co., 101  
11 Cal.App.4th 1443, 1453 (2002) (rejecting assertion that trade  
12 secrets were defined too broadly where, from responses to  
13 discovery, it was clear that former employee had "no  
14 difficulty in understanding the scope of the putative trade  
15 secret information.")

16 Moreover, the scope of trade secrets are specifically  
17 limited to methods developed by Foundstone. (Memo. at 9-11.)  
18 The Court of Appeal has specifically held that an injunction  
19 "need not specify in detail the processes whose use is  
20 prohibited." Components For Research, Inc. v. Isolation  
21 Prods., Inc., 241 Cal.App.2d 726, 731 (1966). Instead, it is  
22

23 Foundstone has provided pages of detail describing the trade  
24 secrets at issue. Defendants' reliance on MAI Sys. Corp. v.  
25 Peak Computer, 991 F.2d 511 (9th Cir. 1993), is also  
26 misplaced. In MAI, the plaintiff cryptically referred only to  
27 "valuable trade secrets" in software, without any further  
28 explanation whatsoever.

<sup>2</sup> Defendants also assert that Foundstone has not complied  
with C.C.P. § 2019(d). This, too, is incorrect. Foundstone  
served a disclosure pursuant to Section 2019(d) on October 15,  
2002.



1 sufficient to describe the technology and specifically limit  
2 the scope of an injunction to "method[s], technique[s], or  
3 processes . . . developed by the plaintiff." Id. (Emphasis  
4 added.) For the additional reason that the trade secrets  
5 claimed here are all specifically limited to methods developed  
6 by Foundstone, the trade secrets are described in more-than-  
7 sufficient detail.

8 (b) The Trade Secrets Are Not Public Domain

9 Defendants also argue that Foundstone's trade secrets are  
10 "based on information available in the public domain." (Opp.  
11 at 10.) This assertion is utterly irrelevant. Foundstone  
12 does not dispute that its trade secrets rely, in part, on  
13 known principles. Defendants, however, never claim that any  
14 of the publications or software they refer to in their papers  
15 discloses the actual methods Foundstone developed.<sup>3</sup> They do  
16 not dispute that the massive investment Foundstone had to make  
17 to create its software. (McClure ¶ 4.) Indeed, Defendants  
18 admit that, to duplicate the functionality of Foundstone's  
19 software, it would be necessary for a competitor to make a  
20 similar investment of money and effort. (Glaser ¶ 31.)

21 (c) Reasonable Efforts To Maintain Secrecy

22 Defendants also assert that Foundstone did not take  
23 reasonable efforts to protect the secrecy of the software.  
24

25 <sup>3</sup> The 152-page Exh. F to the Glaser Decl. is typically  
26 irrelevant to the trade secrets in this case. Defendants  
27 suggest that this document somehow discloses Foundstone's  
28 proprietary TCP and ICMP methods of operating system  
identification. The document, however, merely discusses  
general ICMP methods of network scanning, not the techniques  
developed by Foundstone.

1 (Opp. at 10.) Defendants do not dispute the comprehensive  
2 steps Foundstone has taken to protect its trade secrets.  
3 (McClure Decl. ¶¶ 13-20.) Instead, Defendants point only to  
4 Foundstone's alleged provision of evaluation copies of its  
5 software to reviewers, which would have theoretically allowed  
6 a reviewer to "read, write, or print" a single database.  
7 (Caso Decl. ¶ 15.) Defendants never claim that the data - an  
8 unintelligible mixed series of binary, decimal, and  
9 hexadecimal numbers - could be used or even understood by  
10 anyone. Defendants never state that any source code, or any  
11 means by which anyone could use or understand the allegedly  
12 compromised database was disclosed. Accordingly, there is no  
13 serious dispute that Foundstone took reasonable steps to  
14 protect its trade secrets.

15 2. Defendants Have Misappropriated And Threatened To  
16 Misappropriate Foundstone's Trade Secrets

17 In its moving papers, Foundstone submitted detailed  
18 evidence establishing that Defendants had misappropriated and  
19 threatened to misappropriate its trade secrets, citing to  
20 specific features of Defendants' software. (McClure Decl. ¶¶  
21 30-32.)<sup>4</sup> In their Opposition, Defendants do not dispute  
22 (indeed, they admit) that their software contains these  
23 features. (Caso Decl. Exh. C; Glaser Exh. E.) Moreover,  
24

25  
26 <sup>4</sup> Although Defendants assert objections to paragraphs 30-  
27 32 of the McClure Decl., they do not dispute the information  
28 presented. Moreover, the statements in paragraphs 30-32 of  
the McClure Decl. are confirmed by Defendants' own  
descriptions of their software. (Caso Decl. Exh. C; Glaser  
Decl. Exh. E.)

## **EXHIBIT 'F'**

DESIGNATED LITIGATION ATTORNEY'S COPY  
ORIGINAL IN DOCKETING SYSTEM

Darrell L. Olson, Bar No. 77,633  
Michael K. Friedland, Bar No. 157,217  
Douglas T. Hudson, Bar No. 210,385  
KNOBBE, MARTENS, OLSON & BEAR, LLP  
2040 Main Street  
Fourteenth Floor  
Irvine, California 92614  
(949) 760-0404 (telephone)  
(949) 760-9502 (facsimile)  
Attorneys for Plaintiff FOUNDSTONE, INC.

SUPERIOR COURT OF THE STATE OF CALIFORNIA  
COUNTY OF ORANGE, CENTRAL JUSTICE CENTER

|  |  |
|--|--|
| FOUNDSTONE, INC., a Delaware corporation,  | ) CASE NO. 02CC15350                   |
| Plaintiff,   | ) ASSIGNED FOR ALL PURPOSES TO:        |
|  | ) THE HON. DAVID H. BRICKNER           |
|  | ) DEPT. C17                            |
| v.   | ) PLAINTIFF FOUNDSTONE'S               |
| NT Objectives, Inc., a California corporation; J.D. Glaser, an individual; Michael Morton, an individual; Eric Caso, an individual; Dan Kuykendall, an individual; and DOES 1 through 50, inclusive, | ) <u>SUPPLEMENTAL</u> REPLY IN SUPPORT |
|  | ) OF ORDER TO SHOW CAUSE RE:           |
|  | ) PRELIMINARY INJUNCTION               |
| Defendants.  | ) HEARING DATE: Oct. 25, 2002          |
|  | ) TIME: 3:00 PM                        |
|  | ) DEPT: C17                            |
|  | ) TRIAL DATE: None                     |
|  | ) COMPLAINT FILED: Oct. 2, 2002        |

1 Plaintiff Foundstone, Inc. ("Foundstone") submits this  
2 Supplemental Reply in Support of the Order to Show Cause Re:  
3 Preliminary Injunction. The information addressed in this  
4 Supplemental Reply disproves a key assertion made by  
5 Defendants in their Opposition. The information was not  
6 discovered by Foundstone until after Foundstone filed its  
7 Reply on October 25, 2002.

8 In their Opposition, Defendants repeatedly argue that it  
9 is "impossible" for the Defendants to determine the trade  
10 secrets that are the subject of this action. (Opp. at 2. See  
11 also id. at 6-9.) The Defendants themselves submitted  
12 declarations, each asserting, in lock-step: "I have read the  
13 Declaration of Stuart McClure submitted in support of this OSC  
14 re: Preliminary Injunction. From review of that document I am  
15 unable to determine any specific method, algorithm or  
16 technique which is being referred to by Foundstone as a trade  
17 secret." (Caso Decl. ¶ 13; Glaser Decl. ¶ 26; Kuykendall  
18 Decl. ¶ 15; Morton Decl. ¶ 16.)

19 Information just uncovered by Foundstone demonstrates  
20 that these assertions are false. As set forth in Exh. A to  
21 the Hudson Decl. submitted herewith, Defendant Kuykendall  
22 maintains a personal site on the World Wide Web. On his web  
23 site, Defendant Kuykendall published a paragraph-by-paragraph  
24 critique of the McClure Decl., specifically addressed each of  
25 the paragraphs of the McClure Decl. describing the trade  
26 secrets at issue in this action, and unambiguously admitted  
27 that he understood that Foundstone has proprietary technology  
28 in each described area. The McClure Decl. described the trade

1 secrets at issue in paragraphs 7-12. In his web site,  
2 Defendant Kuykendall responded to each of these paragraphs by  
3 admitting, inter alia, "they [Foundstone] have their  
4 proprietary implementation." (Hudson Decl. Ex. A p. 3, ¶¶ 7-  
5 12; emphasis added.)

6 For this additional reason, Foundstone requests that this  
7 Court preliminarily enjoin Defendants and maintain the status  
8 quo pending final adjudication of this matter.

9 Respectfully Submitted,

10 KNOBBE, MARTENS, OLSON & BEAR, LLP

11  
12 Dated: 10/28/2002

13 By: 

14 Darrell L. Olson  
15 Michael K. Friedland  
16 Douglas T. Hudson  
17 Attorneys for Plaintiff  
18 FOUNDSTONE, INC.  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

ICMP Usage in Scanning  
Version 2.5

# **ICMP Usage in Scanning**

Or

## **Understanding some of the ICMP Protocol's Hazards**

**Ofir Arkin**

**Founder**

**The Sys-Security Group**



<http://www.sys-security.com>  
[ofir@sys-security.com](mailto:ofir@sys-security.com)

**Version 2.5**

**December 2000**

1

Copyright © Ofir Arkin, 2000-2001  
<http://www.sys-security.com>

ICMP Usage In Scanning  
Version 2.5

## Table of Contents

|  |    |
|--|----|
| Introduction.....  | 7  |
| 1.1 Introduction to Version 1.0 .....  | 7  |
| 1.2 Introduction to Version 2.0 .....  | 7  |
| 1.3 Introduction to Version 2.5 .....  | 8  |
| 1.4 CHANGES .....  | 8  |
| 1.4.1 Version 1.0 to Version 2.0 .....   | 8  |
| 1.4.2 Version 2.0 to Version 2.01 .....  | 9  |
| 1.4.3 Version 2.01 to Version 2.5 .....  | 9  |
| 2.1 ICMP ECHO (Type 8) and ECHO Reply (Type 0) .....   | 10 |
| 2.2 ICMP Sweep (Ping Sweep).....   | 11 |
| 2.3 Broadcast ICMP.....  | 12 |
| 2.4 Non-ECHO ICMP.....   | 14 |
| 2.4.1 ICMP Time Stamp Request (Type 13) and Reply (Type 14).....   | 15 |
| 2.4.2 ICMP Information Request (Type 15) and Reply (Type 16).....  | 16 |
| 2.4.3 ICMP Address Mask Request (Type 17) and Reply (Type 18).....   | 19 |
| 2.5 Non-ECHO ICMP Sweeps .....   | 22 |
| 2.6 Non-ECHO ICMP Broadcasts .....   | 23 |
| 3.0 Advanced Host Detection using the ICMP Protocol (using ICMP Error Messages<br>generated from the probed machines).....   | 25 |
| 3.1 Sending IP Datagrams with bad IP headers fields – generating ICMP Parameter<br>Problem error message back from probed machines.....                              | 25 |
| 3.1.1 ACL Detection using IP Datagrams with bad IP headers fields .....  | 27 |
| 3.2 IP Datagrams with non-valid field values.....  | 29 |
| 3.2.1 The Protocol Field example.....  | 29 |
| 3.2.1.2 Using all combination of the IP protocol filed values.....   | 29 |
| 3.2.2 ACL Detection using the Protocol field .....   | 30 |
| 3.3 Host Detection using IP fragmentation to elicit Fragment Reassembly Time<br>Exceeded ICMP error message.....   | 31 |
| 3.3.1 ACL Detection using IP fragmentation .....   | 32 |
| 3.4 Host Detection using UDP Scans, or why we wait for the ICMP Port Unreachable<br>.....  | 33 |
| 3.4.1 A Better Host Detection Using UDP Scan .....   | 34 |
| 3.5 Using Packets bigger than the PMTU of internal routers to elicit an ICMP<br>Fragmentation Needed and Don't Fragment Bit was Set (configuration problem)<br>..... | 35 |
| 4.0 Inverse Mapping .....  | 36 |
| 4.1 Inverse Mapping Using ICMP (Echo & Echo Reply).....  | 36 |
| 4.2 Inverse Mapping Using Other Protocols.....   | 37 |
| 4.3 Patterns we might see .....  | 37 |
| 5.0 Using traceroute to Map a Network Topology .....   | 40 |
| 6.0 The usage of ICMP in Active Operating System Fingerprinting Process .....  | 43 |
| Using Regular ICMP Query Messages .....  | 43 |
| 6.1.2 Using ICMP Information Requests .....  | 44 |
| 6.1.3 Identifying Operating Systems according to their replies for non-ECHO ICMP<br>requests aimed at the broadcast address.....                                     | 44 |
| 6.2 The DF Bit Playground (Identifying Sun Solaris, HP-UX 10.30, 11.0x, and AIX<br>4.3.x based machines).....  | 45 |



ICMP Usage In Scanning  
Version 2.5

|  |     |
|--|-----|
| 6.2.1 Avoidance .....  | 52  |
| 6.3 The IP Time-to-Live Field Value with ICMP .....  | 53  |
| 6.3.1 IP TTL Field Value with ICMP Query Replies .....   | 53  |
| 6.3.2 IP TTL Field Value with ICMP ECHO Requests .....   | 55  |
| 6.3.3 Correlating the Information .....  | 56  |
| 6.4 Using Fragmented ICMP Address Mask Requests (Identifying Sun Solaris & HP-UX 11.0x machines) .....                 | 57  |
| Using Crafted ICMP Query Messages .....  | 59  |
| Playing with the TOS Field .....   | 59  |
| 6.5 Precedence Bits Echoing (Fingerprinting Microsoft Windows 2000, ULTRIX, HPUX 11.0&10.30, OpenVMS and more) .....   | 61  |
| 6.5.1 Changed Pattern with other ICMP Query Message Types .....  | 67  |
| 6.6 TOSing OSs out of the Window / "TOS Echoing" (Fingerprinting Microsoft Windows 2000) .....                         | 69  |
| 6.6.1 The use of the Type-of-Service field with the Internet Control Message Protocol .....                            | 69  |
| 6.7 Using the TOS Byte's Unused Bit (Fingerprinting Microsoft Windows 2000, ULTRIX and more) .....                     | 75  |
| 6.7.1 Changed Pattern with Replies for Different ICMP Query Types .....  | 77  |
| 6.8 Using the Unused (Identifying Sun Solaris & HP-UX 10.30 & 11.0x OS based machines) .....                           | 78  |
| 6.9 DF Bit Echoing .....   | 80  |
| 6.9.1 DF Bit Echoing with the ICMP Echo request .....  | 80  |
| 6.9.2 DF Bit Echoing with the ICMP Address Mask request .....  | 81  |
| 6.9.3 DF Bit Echoing with the ICMP Timestamp request .....   | 81  |
| 6.9.4 Using all of the Information in order to identify maximum of operating systems .....                             | 82  |
| 6.9.5 Why this would work (for the skeptical) .....  | 82  |
| 6.9.6 Combining all together .....   | 83  |
| 6.10 Using Code field values different than zero within ICMP ECHO requests .....                                       | 85  |
| 6.11 Using Code field values different than zero within ICMP Timestamp Request .....                                   | 86  |
| 6.11.1 The non-answering Operating Systems .....   | 86  |
| 6.11.2 Operating Systems the Zero out the Code field value on Reply .....  | 87  |
| Using the ICMP Error Messages .....  | 89  |
| 6.12 Operating system, which do not generate ICMP Protocol Unreachable Error Messages .....                            | 89  |
| 6.13 ICMP Error Message Quenching .....  | 90  |
| 6.14 ICMP Error Message Quoting Size .....   | 90  |
| 6.15 LINUX ICMP Error Message Quoting Size Differences / The 20 Bytes from No Where .....                              | 92  |
| 6.16 Foundry Networks Networking Devices Padded Bytes with ICMP Port Unreachable(s) / The 12 Bytes from No Where ..... | 94  |
| 6.17 ICMP Error Message Echoing Integrity (Tested with ICMP Port Unreachable) .....                                    | 95  |
| 6.18 Novell Netware Echoing Integrity Bug with ICMP Fragment Reassembly Time Exceeded .....                            | 100 |
| 6.19 The Precedence bits with ICMP Error Messages (Identifying LINUX) .....  | 101 |
| 6.20 TOS Bits (-field) Echoing with ICMP Error .....   | 104 |
| 6.21 DF Bit Echoing with ICMP Error Messages .....   | 105 |
| Not that useful fingerprinting method(s) .....   | 112 |
| 6.22 Unusual Big ICMP Echo Request .....   | 112 |
| 7.0 Filtering ICMP on your Filtering Device to Prevent Scanning Using ICMP .....                                       | 114 |

ICMP Usage in Scanning  
Version 2.5

|  |     |
|--|-----|
| 7.1 Inbound.....   | 114 |
| 7.2 Outbound.....  | 114 |
| 7.3 Other Considerations.....  | 116 |
| 7.4 Other Problems – Why it is important to filter ICMP traffic in the Internal<br>segmentation .....                  | 117 |
| 7.5 The Firewall.....  | 118 |
| 8.0 Conclusion.....  | 120 |
| 9.0 Acknowledgment.....  | 121 |
| 9.1 Acknowledgment for version 1.0 .....   | 121 |
| 9.1 Acknowledgment for version 2.0 .....   | 121 |
| 9.2 Acknowledgment for version 2.5.....  | 121 |
| Appendix A: The ICMP Protocol .....  | 122 |
| A.1 ICMP Messages.....   | 123 |
| A.1 ICMP Error Messages .....  | 125 |
| A.1.1 ICMP Error Messages .....  | 126 |
| A.1.1.1 Destination Unreachable (Type 3) .....   | 126 |
| A.1.1.2 Source Quench (Type 4).....  | 128 |
| A.1.1.3 Redirect (Type 5).....   | 129 |
| A.1.1.4 Time Exceeded (Type 11).....   | 130 |
| A.1.1.5 Parameter Problem (Type 12).....   | 130 |
| Appendix B: ICMP "Fragmentation Needed but the Don't Fragment Bit was set" and the<br>Path MTU Discovery Process ..... | 132 |
| B.1 The PATH MTU Discovery Process.....  | 132 |
| B.2 Host specification .....   | 132 |
| B.3 Router Specification .....   | 133 |
| B.4 The TCP MSS (Maximum Segment Size) Option and PATH MTU Discovery<br>Process.....                                   | 134 |
| Appendix C: Mapping Operating Systems for answering/discarding ICMP query<br>message types.....                        | 135 |
| Appendix D: ICMP Query Message Types with Code field !=0 .....   | 137 |
| Appendix E: ICMP Query Message Types aimed at a Broadcast Address.....   | 139 |
| Appendix F: Precedence Bits Echoing with ICMP Query Request & Reply.....   | 141 |
| Appendix G: ICMP Query Message Types with TOS! = 0.....  | 142 |
| Appendix H: Echoing the TOS Byte Unused bit .....  | 143 |
| Appendix I: Using the Unused Bit .....   | 144 |
| Appendix J: DF Bit Echoing .....   | 145 |
| Appendix K: ICMP Error Message Echoing Integrity with ICMP Port Unreachable Error<br>Message .....                     | 146 |
| Appendix L: Snort Basic Rule Base for ICMP Traffic .....   | 148 |

# ICMP Usage in Scanning Version 2.5

## Figures List

|  |     |
|--|-----|
| Figure 1: ICMP ECHO Mechanism  | 10  |
| Figure 2: ICMP ECHO Request & Reply message format   | 11  |
| Figure 3: Broadcast ICMP   | 13  |
| Figure 4: ICMP Time Stamp Request & Reply message format   | 15  |
| Figure 5: ICMP Information Request and Reply   | 17  |
| Figure 6: ICMP Address Mask Request & Reply message format   | 19  |
| Figure 7: The IP Header  | 25  |
| Figure 8: An Example: A TCP packet fragmented after only 8 bytes of TCP information  | 33  |
| Figure 9: Using Packets bigger than the PMTU of Internal routers to elicit an ICMP Fragmentation Needed and Don't Fragment Bit was Set | 35  |
| Figure 10: ICMP Time Exceeded message format   | 40  |
| Figure 11: The Type of Service Byte  | 59  |
| Figure 12: ICMP ECHO Request & Reply message format  | 86  |
| Figure 13: The Type of Service Byte  | 101 |
| Figure 14: Firewall ICMP Filtering Rules   | 117 |
| Figure 15: Internal segmentation ICMP Filtering Example  | 118 |
| Figure 16: ICMP Message Format   | 123 |
| Figure 17: ICMP Error Message General Format   | 126 |
| Figure 18: ICMP Fragmentation Needed but the Don't Fragment Bit was set Message Format   | 128 |
| Figure 19: ICMP Redirect Message Format  | 129 |
| Figure 20: ICMP Parameter Problem Message Format   | 131 |
| Figure 21: ICMP Fragmentation Required with Link MTU   | 133 |

## Table List

|  |     |
|--|-----|
| Table 1: Which Operating System would answer to an ICMP ECHO Request aimed at the Broadcast Address of the Network they reside on?               | 14  |
| Table 2: Non-ECHO ICMP Query of different Operating Systems and Networking Devices   | 22  |
| Table 3: Operating Systems, which would answer to requests, aimed at the Broadcast address   | 24  |
| Table 4: Networking Devices, which would answer to requests, aimed at the Broadcast address  | 24  |
| Table 5: IP TTL Field Values in replies from Various Operating Systems   | 53  |
| Table 6: IP TTL Field Values in requests from Various Operating Systems  | 55  |
| Table 7: Further dividing the groups of operating systems according to IP TTL field value in the ICMP ECHO Requests and in the ICMP ECHO Replies | 56  |
| Table 8: Precedence Field Values   | 60  |
| Table 9: Type-of-Service Field Values  | 60  |
| Table 10: ICMP Query Message Types with Precedence Bits I = 0  | 68  |
| Table 11: ICMP Query Message Types with TOS! = 0   | 75  |
| Table 12: ICMP Query Message Types with the TOS Byte Unused Bit value I = 0  | 78  |
| Table 13: DF Bit Echoing   | 83  |
| Table 14: ICMP Error Message Echoing Integrity   | 98  |
| Table 15: Precedence Field Values  | 102 |
| Table 16: ICMP message types   | 122 |
| Table 17: ICMP Types & Codes   | 124 |
| Table 18: Destination Unreachable Codes (Router)   | 127 |
| Table 19: Redirect Codes   | 129 |
| Table 20: Parameter Problem Codes  | 131 |

## Diagram List

|   |    |
|---|----|
| Diagram 1: The Inverse Mapping Idea   | 37 |
| Diagram 2: A Decoy Scan Example   | 39 |
| Diagram 3: Finger Printing Using ICMP Information Request Combines with ICMP Address Mask Request | 44 |

ICMP Usage in Scanning  
Version 2.5

|  |     |
|--|-----|
| Diagram 4: Finger Printing Using non-ECHO ICMP Query Types aimed at the Broadcast Address of an Attacked Network   | 45  |
| Diagram 5: Finger Printing Using ICMP Address Mask Requests  | 59  |
| Diagram 6: An example for a way to fingerprint Microsoft Windows 2000, Ultrix, HP-UX 11.0 & 10.30, OpenVMS, Microsoft Windows ME, and Microsoft Windows 98/98SE based machines with ICMP Query messages with the Precedence Bits field I=0 | 68  |
| Diagram 7: An example for a way to fingerprint Windows 2000, Ultrix, and Novell Netware based machines with ICMP Query messages with the TOS bits field I=0  | 74  |
| Diagram 8: An example for a way to fingerprint operating systems using the unused bit in the TOS Byte echoing method   | 77  |
| Diagram 9: An example of fingerprinting using the DF Bit Echoing technique   | 84  |
| Diagram 10: Finger Printing Using ICMP Timestamp Request and Wrong Codes   | 87  |
| Diagram 11: An Example of Finger Printing Using crafted ICMP Echo & Timestamp Request  | 89  |
| Diagram 12: DF Bit Echoing with ICMP Error Messages  | 107 |

ICMP Usage in Scanning  
Version 2.5

## Introduction

### 1.1 Introduction to Version 1.0

The ICMP Protocol may seem harmless at first glance. Its goals and features were outlined in RFC 792 (and then later cleared in RFCs 1122, 1256, 1349, 1812), as a way to provide a means to send error messages. In terms of security, ICMP is one of the most controversial protocols in the TCP/IP protocol suite. The risks involved in implementing the ICMP protocol in a network, regarding scanning, are the subject of this research paper.

Scanning will usually be the major stage of an information gathering process a malicious computer attacker will launch against a targeted network. With this stage the malicious computer attacker will try to determine what are the characteristics of the targeted network. He will use several techniques, such as host detection, service detection, network topology mapping, and operating system fingerprinting. The data collected will be used to identify those hosts (if any) that are running a network service, which may have a known vulnerability. This vulnerability may allow the malicious computer attacker to execute a remote exploit in order to gain unauthorized access to those systems. This unauthorized access may become his focal point to the whole targeted network.

This research paper outlines the usage of the ICMP protocol in the scanning process. Step-by-step we will uncover each of the malicious computer attacker techniques using the ICMP protocol. A few new scanning techniques will be unveiled in this research paper. I have reported some of them to several security mailing lists, including Bugtraq, in the past.

The chapters in this research paper are divided according to the various scanning techniques:

- Host Detection using the ICMP protocol is dealt in Chapter 2.
- Advanced Host Detection methods using the ICMP protocol are discussed in Chapter 3.
- Inverse Mapping using the ICMP protocol is discussed in Chapter 4.
- Network Mapping using the *traceroute* utility is discussed in Chapter 5.
- Chapter 6 discusses the usage of ICMP in the Active Operating System Fingerprinting process.
- Chapter 7 suggests a filtering policy to be used on filtering devices when dealing with the ICMP protocol.

I would like to take a stand in this controversial issue. ICMP protocol hazards are not widely known. I hope this research paper will change this fact.

### 1.2 Introduction to Version 2.0

Quite a large number of new OS fingerprinting methods using the ICMP protocol, which I have found are introduced with this revision. Among those methods two can be used in order to identify Microsoft Windows 2000 machines; one would allow us to distinguish between Microsoft Windows operating system machines and the rest of the world, and another would allow us to distinguish between SUN Solaris machines and the rest of the world<sup>1</sup>. I have also tried to be accurate as possible with data presented in this paper. Few tables have been added to the paper mapping the behavior of the various operating systems I have used. These tables describe the results I got from the various machines after querying them with the various tests introduced with this paper.

See section 1.3 for a full Changes list.

---

<sup>1</sup> See Section 6 for more information.

ICMP Usage in Scanning  
Version 2.5

### 1.3 Introduction to Version 2.5

With this version of the research paper I am introducing a few new OS fingerprinting methods. Some are targeted in producing ICMP error messages from a target OS, enabling us to fingerprint an OS even if all ports of the OS in question are closed. I have also added a considerable amount of information about ICMP error message. At the end of the paper you will find the Basic snort rule base I have written.

## 1.4 CHANGES

### 1.4.1 Version 1.0 to Version 2.0

#### 2.0 Host Detection Using the ICMP Protocol

##### 2.3 Broadcast ICMP

Added a table describing which operating systems would answer an ICMP ECHO request aimed at the Broadcast address of the network they reside on.

##### 2.4 Non-ECHO ICMP

Added Information Request and Reply as a valid Host Detection method.

##### 2.4.2 ICMP Information Request and Reply

The actual Information (added a section).

##### 2.4.3 ICMP Address Mask Request and Reply

Added SUN Solaris and networking devices examples.

##### 2.5 Non-Echo ICMP Sweep

Added a table summarizing which operating systems would answer those queries.

##### 2.6 Non-ECHO ICMP Broadcasts

Added the fact that "Hosts running an operating system, which answers requests aimed at the IP broadcast address..."

Added two tables describing which operating systems would answer to which type of ICMP queries aimed at the broadcast address of the network they reside on?

#### 3.0 Host Detection Using ICMP Error messages generated from the probed machines

##### 3.1 IP datagrams with bad IP Header fields

Added more information on various other fields which can be used for this purpose.

#### 6.0 The Usage of ICMP in the operating system Finger Printing Process

##### 6.1 Using Wrong Codes within ICMP Datagrams

6.1.1 Using ICMP Timestamp Requests with Codes different than 0

6.1.2 Listing ICMP query message types sent to different operating systems with the Code field !=0 and the answers (if any) we got.

##### 6.2 Using ICMP Address Mask Requests (Identifying Solaris Machines)

##### 6.3 TOSing OSs out of the Window / Fingerprinting Microsoft Windows 2000

##### 6.7 Using ICMP Address Mask Requests

##### 6.8 Using ICMP Information Requests

6.9 Identifying operating systems according to their replies for non-ECHO ICMP requests aimed at the broadcast address.

##### 6.10 IP TTL Field Value with ICMP

6.10.1 IP TTL Field Value with ICMP ECHO Replies

6.10.2 IP TTL Field Value with ICMP ECHO Requests

ICMP Usage in Scanning  
Version 2.5

6.11 DF Bit

6.12 DF Bit Echoing

6.12.1 DF Bit Echoing with ICMP Echo requests

6.12.2 DF Bit Echoing with ICMP Address Mask requests

6.12.3 DF Bit Echoing with ICMP Timestamp requests

6.12.4 Using all of the information in order to identify the maximum of operating systems.

6.12.5 Why this would work (for the skeptical)

6.13 What will not provide any gain compared to the effort and the detection ability?

6.13.1 Unusual big ICMP Echo messages

7.0 Filtering ICMP on your Filtering Device to Prevent Scanning Using ICMP

7.3 Other Considerations

More information was added.

Appendixes

Appendix C: Table - Mapping Operating Systems for answering/discarding ICMP query Message types.

Appendix D: Table - ICMP Query Message Types with Code Field !=0

Appendix E: Table - ICMP Query Message Types aimed at a Broadcast Address

Appendix F: Table - ICMP Query Message Types with TOS !=0

Appendix G: Table - DF Bit Echoing

1.4.2 Version 2.0 to Version 2.01

The Introduction was re-written

1.4.3 Version 2.01 to Version 2.5

To Section 4 "Inverse Mapping" more information and explanations were added.

Section 6 is now divided into four main subjects:

- Fingerprinting using regular ICMP Query requests
- Fingerprinting using crafted ICMP Query request
- Fingerprinting using ICMP Error Messages
- Not that useful fingerprinting methods

Multiple new fingerprinting methods based on ICMP Error Messages were introduced.

I have also introduced few Fingerprinting method based on ICMP Query messages: "Using the Unused (Identifying Sun Solaris & HP-UX 10.30 & 11.0x)", "Precedence Bits Echoing (Win2k, ULTRIX Identification)", "The TOS Byte Unused Bit Echoing (Identifying Win2k, ULTRIX)".

"The DF Bit Playground" fingerprinting method was better explained and explored.

Appendix A now includes explanation for the various ICMP Error Messages.

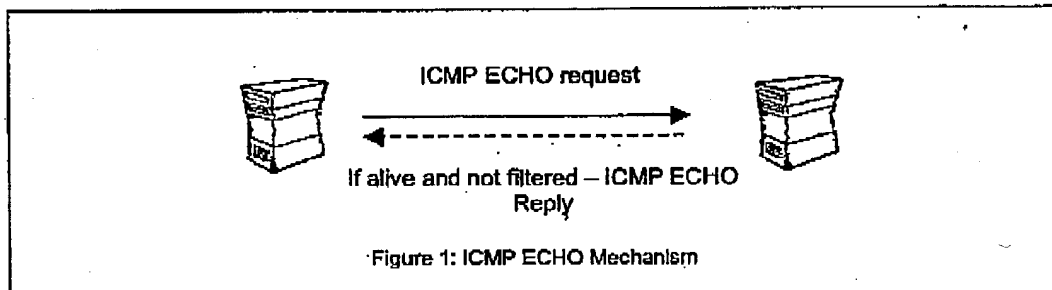
# ICMP Usage In Scanning Version 2.5

## 2.0 Host Detection using the ICMP Protocol<sup>2</sup>

The Host Detection stage gives a malicious computer attacker crucial information by identifying the computers on the targeted network that are reachable from the Internet. This process belongs to the scanning stage, which is one of the first stages in the Information Gathering process. The information collected during this stage could later lead to an attempt to break in to one (or more) of the targeted network computers. This, if the information gathered would be sufficient for the malicious computer attacker.

## 2.1 ICMP ECHO (Type 8) and ECHO Reply (Type 0)

We can use an *ICMP ECHO* datagram to determine whether a target IP address is active or not, by simply sending an *ICMP ECHO*<sup>3</sup> (ICMP type 8) datagram to the targeted system and waiting to see if an *ICMP ECHO Reply* (ICMP type 0) is received. If an *ICMP ECHO* reply is received, it would indicate that the target is alive (few firewalls spoof *ICMP ECHO* replies from protected hosts); No response means the target is down or a filtering device is preventing the incoming *ICMP ECHO* datagram from getting inside the protected network or the filtering device prevents the initiated reply from reaching the Internet.



This mechanism is used by the Ping command to determine if a destination host is reachable.

In the next example two LINUX machines demonstrate the usage of Ping:

```
[root@stan /root]# ping 192.168.5.5
PING 192.168.5.5 (192.168.5.5) from 192.168.5.1 : 56(84) bytes of data.
64 bytes from 192.168.5.5: icmp_seq=0 ttl=255 time=4.4 ms
64 bytes from 192.168.5.5: icmp_seq=1 ttl=255 time=5.9 ms
64 bytes from 192.168.5.5: icmp_seq=2 ttl=255 time=5.8 ms

--- 192.168.5.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 4.4/5.3/5.9 ms
```

### A Short trace<sup>4</sup>:

```
01/26-13:16:25.746316 192.168.5.1 -> 192.168.5.5
```

<sup>2</sup> For more information about the ICMP Protocol please read "Appendix A: The ICMP Protocol".

<sup>3</sup> From a technical point of view: The sending side initializes the identifier (used to identify ECHO requests aimed at different destination hosts) and sequence number (if multiple ECHO requests are sent to the same destination host), adds some data (arbitrary) to the data field and sends the *ICMP ECHO* to the destination host. In the ICMP header the code equals zero. The recipient should *only change* the type to *ECHO Reply* and return the datagram to the sender.

<sup>4</sup> Snort, written by Martin Roesch, can be found at <http://www.snort.org>.



# ICMP Usage in Scanning Version 2.5

ICMP TTL:64 TOS:0x0 ID:6059

ID:5721 Seq:1 ECHO

```

89 D7 8E 38 27 63 0B 00 08 09 0A 0B 0C 0D 0E 0F ...8'c.....
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F .....
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 1"#$%&'()*+,-./
30 31 32 33 34 35 36 37 01234567

```

01/26-13:16:25.746638 192.168.5.5 -&gt; 192.168.5.1

ICMP TTL:255 TOS:0x0 ID:6072

ID:5721 Seq:1 ECHO REPLY

```

89 D7 8E 38 27 63 0B 00 08 09 0A 0B 0C 0D 0E 0F ...8'c.....
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F .....
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 1"#$%&'()*+,-./
30 31 32 33 34 35 36 37 01234567

```

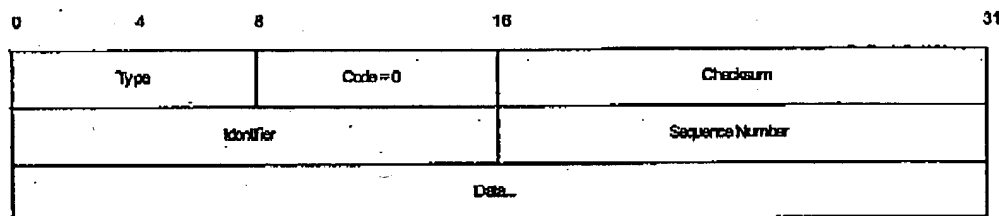


Figure 2: ICMP ECHO Request &amp; Reply message format

**Countermeasure:** Block ICMP ECHO requests coming from the Internet towards your network at your border router and/or Firewall<sup>5</sup>.

## 2.2 ICMP Sweep (Ping Sweep)

Querying multiple hosts using ICMP ECHO is referred to as *ICMP Sweep* (or *Ping Sweep*).

For a small to midsize network the Ping utility is an acceptable solution to this kind of host detection, but with large networks (such as Class A, or a full Class B) this kind of scan is fairly slow mainly because Ping waits for a reply (or a time out to be reached) from the probed host before proceeding to the next one.

*fping*<sup>6</sup> is a UNIX utility which sends parallel mass ECHO requests in a round robin fashion enabling it to be significantly faster than the usual Ping utility. It can also be fed with IP addresses with its accompanied tool *gping*. *gping* is used to generate a list of IP addresses which would be later fed into *fping*, directly or from a file, to perform the ICMP sweep. *fping* is also able to resolve hostnames of the probed machines if using the *-d* option.

Another UNIX tool that is able of doing an ICMP sweep in parallel, resolve the hostnames of the probed machines, save it to a file and a lot more is *NMAP*<sup>7</sup>, written by Fyodor.

<sup>5</sup> It is better to filter unwanted traffic at your border router, reducing traffic rates for your firewall.

<sup>6</sup> <http://ftp.tamu.edu/pub/Unix/src>

<sup>7</sup> <http://www.insecure.org>

#### ICMP Usage in Scanning Version 2.5

For the Microsoft Windows operating system a notable ICMP sweep tool is Pinger from Rhino9<sup>8</sup>, able of doing what fping and NMAP do regarding this kind of scan.

Trying to resolve the names of the probed machines may discover the malicious computer attacker's IP number used for the probing, using the log of the authoritative DNS server.

The next example demonstrates the usage of NMAP to perform an ICMP sweep<sup>9</sup> against 20 IP addresses. Our test lab contains two LINUX machines running Redhat Linux v6.1, Kernel 2.2.12 (Stan & Kenny) and one Windows NT WRKS SP4 (Cartman). As it can be seen all of the machines answered the probe:

```
[root@stan /root]# nmap -sP -PI 192.168.5.1-20
```

```
Starting nmap V. 2.3BETA13 by fyodor@insecure.org (
www.insecure.org/nmap/ )
Host stan.sys-security.com (192.168.5.1) appears to be up.
Host kenny.sys-security.com (192.168.5.5) appears to be up.
Host cartman.sys-security.com (192.168.5.15) appears to be up.
Nmap run completed -- 20 IP addresses (3 hosts up) scanned in 3 seconds
```

If we wish to avoid the automatic resolving done by NMAP we should use the `-n` option to eliminate it.

ICMP sweeps are easily detected by IDS (Intrusion Detection Systems) whether launched in the regular way, or if used in a parallel way.

**Countermeasure:** Block ICMP ECHO requests coming from the Internet towards your network at your border router and/or Firewall.

### 2.3 Broadcast ICMP

A simpler way to map a targeted network for alive hosts is by sending an ICMP ECHO request to the broadcast address or to the network address of the targeted network.

The request would be broadcasted to all hosts on the targeted network. The alive hosts will send an ICMP ECHO Reply to the prober's source IP address (additional conditions apply here).

The malicious computer attacker has to send only one IP packet to produce this behavior.

This technique of host detection is applicable only to some of the UNIX and UNIX-like hosts of the targeted network. Microsoft Windows based machines will not generate an answer (ICMP ECHO Reply) to an ICMP ECHO request aimed at the broadcast address or at the network address. They are configured not to answer those queries out-of-the box (This applies to all Microsoft Windows operating systems except for Microsoft Windows NT 4.0 with service pack below SP4). This is not an abnormal behavior as RFC 1122<sup>10</sup> states that if we send an ICMP ECHO request to an IP Broadcast or IP Multicast addresses it *may* be silently discarded by a host.

<sup>8</sup> The Rhino9 group no longer exists. Their tools are available from a number of sites on the Internet.

<sup>9</sup> The `-sP -PI` options enable NMAP to perform only an ICMP Sweep. The default behavior when using the `-sP` option is different and includes the usage of TCP ACK host detection technique as well.

<sup>10</sup> RFC 1122: Requirements for Internet Hosts - Communication Layers, <http://www.ietf.org/rfc/rfc1122.txt>.

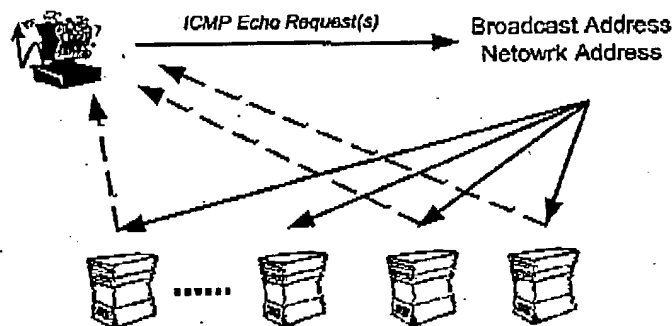
ICMP Usage In Scanning  
Version 2.5

Figure 3: Broadcast ICMP

The next example demonstrates the behavior expected from hosts when sending an ICMP ECHO request to the broadcast address of a network. The two LINUX machines on our test lab answered the query while the Microsoft Windows NT 4.0 Workstation with SP6a machine silently ignored it.

```
[root@stan /root]# ping -b 192.168.5.255
WARNING: pinging broadcast address
PING 192.168.5.255 (192.168.5.255) from 192.168.5.1 : 56(84) bytes of data.
64 bytes from 192.168.5.1: icmp_seq=0 ttl=255 time=4.1 ms
64 bytes from 192.168.5.5: icmp_seq=0 ttl=255 time=5.7 ms (DUP!)

--- 192.168.5.255 ping statistics ---
1 packets transmitted, 1 packets received, +1 duplicates, 0% packet loss
round-trip min/avg/max = 4.1/4.9/5.7 ms
```

In the next example I have sent an ICMP ECHO request to the network address of the targeted network. The same behavior was produced. The LINUX machines answered the ICMP ECHO request while the Microsoft Windows NT 4.0 with SP6a machine ignored it.

```
[root@stan /root]# ping -b 192.168.5.0
WARNING: pinging broadcast address
PING 192.168.5.0 (192.168.5.0) from 192.168.5.1 : 56(84) bytes of data.
64 bytes from 192.168.5.1: icmp_seq=0 ttl=255 time=7.5 ms
64 bytes from 192.168.5.5: icmp_seq=0 ttl=255 time=9.1 ms (DUP!)

--- 192.168.5.0 ping statistics ---
1 packets transmitted, 1 packets received, +1 duplicates, 0% packet loss
round-trip min/avg/max = 7.5/8.3/9.1 ms
```

Note: Broadcast ICMP may result in a *Denial-Of-Service* condition if a lot of machines response to the query at once.

# ICMP Usage in Scanning Version 2.5

A more accurate table that lists which operating systems would answer to an ICMP ECHO request aimed at their Network / Broadcast address is given below:

| Operating System                         | Echo Request<br>Broadcast |
|--|---------------------------|
| Debian GNU/ LINUX 2.2, Kernel 2.4 test 2 | +                         |
| Redhat LINUX 6.2 Kernel 2.2.14           | +                         |
| FreeBSD 4.0                              | -                         |
| FreeBSD 3.4                              | -                         |
| OpenBSD 2.7                              | -                         |
| OpenBSD 2.6                              | -                         |
| NetBSD                                   | -                         |
| Solaris 2.5.1                            | +                         |
| Solaris 2.6                              | +                         |
| Solaris 2.7                              | +                         |
| Solaris 2.8                              | +                         |
| HP-UX v10.20                             | +                         |
| AIX                                      | -                         |
| ULTRIX                                   | -                         |
| Windows 95                               | -                         |
| Windows 98                               | -                         |
| Windows 98 SE                            | -                         |
| Windows ME                               | -                         |
| Windows NT 4 WRKS SP 3                   | -                         |
| Windows NT 4 WRKS SP 6a                  | -                         |
| Windows NT 4 Server SP4                  | -                         |
| Windows 2000 Professional (and SP1)      | -                         |
| Windows 2000 Server (and SP1)            | -                         |

Table 1: Which Operating Systems would answer to an ICMP ECHO Request aimed at the Broadcast Address of the Network they reside on?

**Countermeasure:** Block the IP directed broadcast on the border router.

## 2.4 Non-ECHO ICMP

ICMP ECHO is not the only ICMP query message type available with the ICMP protocol.

Non-ECHO ICMP messages are being used for more advanced ICMP scanning techniques (not only probing hosts, but network devices, such as a router, as well).

The group of ICMP query message types includes the following:

- ECHO Request (Type 8), and Reply (Type 0)
- Time Stamp Request (Type 13), and Reply (Type 14)
- Information Request (Type 15), and Reply (Type 16)
- Address Mask Request (Type 17), and Reply (Type 18)
- Router Solicitation (Type 10), and Router Advertisement (Type 9)

# ICMP Usage in Scanning Version 2.5

## 2.4.1 ICMP Time Stamp Request (Type 13) and Reply (Type 14)

The *ICMP Time Stamp Request and Reply* allows a node to query another for the current time. This allows a sender to determine the amount of latency that a particular network is experiencing. The sender initializes the identifier (used to identify Timestamp requests aimed at different destination hosts) and sequence number (if multiple Timestamp requests are sent to the same destination host), sets the originate time stamp and sends it to the recipient.

The receiving host fills in the receive and transmit time stamps, change the type of the message to time stamp reply and returns it to the recipient. The time stamp is the number of milliseconds elapsed since midnight UT (GMT).

The originate time stamp is the time the sender last touched the message before sending it, the receive time stamp is the time the recipient first touched it on receipt, and the Transmit time stamp is the time the receiver last touched the message on sending it.

|                     |      |                 |    |    |
|---------------------|------|-----------------|----|----|
| 0                   | 4    | 8               | 16 | 31 |
| Type                | Code | Checksum        |    |    |
| Identifier          |      | Sequence Number |    |    |
| Originate timestamp |      |                 |    |    |
| Receive timestamp   |      |                 |    |    |
| Transmit timestamp  |      |                 |    |    |

Figure 4: ICMP Time Stamp Request & Reply message format

As RFC 1122 state, a *host may* implement Timestamp and Timestamp Reply. If they are implemented a host must follow this rules:

- o Minimum variability delay in handling the Timestamp request.
- o The receiving host *must* answer to every Timestamp request that he receives.
- o An ICMP Timestamp Request to an IP Broadcast or IP Multicast address *may* be silently discarded.
- o The IP source address in an ICMP Timestamp reply *must* be the same as the specific-destination address of the corresponding Timestamp request message.
- o If a source-route option is received in a Timestamp request, the return route *must* be reserved and used as a Source Route option for the Timestamp Reply option.
- o If a Record Route and/or Timestamp option is received in a Timestamp request, this option(s) *should* be updated to include the current host and included in the IP header of the Timestamp Reply message.

Receiving an ICMP Timestamp Reply would reveal an alive host (or a networking device) that has implemented the ICMP Timestamp messages.

# ICMP Usage in Scanning Version 2.5

In the next example I have sent an ICMP Time Stamp Request, using the `icmpush`<sup>11</sup> tool, to a Redhat 6.1 LINUX, Kernel 2.2.12 machine:

```
[root@stan /root]# icmpush -tstamp 192.168.5.5
kenny.sys-security.com -> 13:48:07
```

## Snort Trace:

```
01/26-13:51:29.342647 192.168.5.1 -> 192.168.5.5
ICMP TTL:254 TOS:0x0 ID:13170
TIMESTAMP REQUEST
88 16 D8 D9 02 8B 63 3D 00 00 00 00 00 00 00 00 .....C=.....

01/26-13:51:29.342885 192.168.5.5 -> 192.168.5.1
ICMP TTL:255 TOS:0x0 ID:6096
TIMESTAMP REPLY
88 16 D8 D9 02 8B 63 3D 02 88 50 18 02 88 50 18 .....C=...P...P.
2A DE 1C 00 A0 F9 *.....
```

When I have sent an ICMP Time Stamp Request to a Windows NT WRKS 4.0 SP4 machine, I got no reply. Again, this is not an abnormal behavior from the Microsoft Windows NT machine, just an implementation choice as RFC 1122 states.

**Countermeasure:** Block ICMP Time Stamp Requests coming from the Internet on the border Router and/or Firewall.

## 2.4.2 ICMP Information Request (Type 15) and Reply (Type 16)

The *ICMP Information Request/Reply* pair was intended to support self-configuring systems such as diskless workstations at boot time, to allow them to discover their network address.

The sender fills in the request with the Destination IP address in the IP Header set to zero (meaning this network). The request may be sent with both Source IP Address and Destination IP Address set to zero. The sender initializes the identifier and the sequence number, both used to match the replies with the requests, and sends out the request. The ICMP header code field is zero.

If the request was issued with a non-zero Source IP Address the reply would only contain the network address in the Source IP Address of the reply. If the request had both the Source IP Address and the Destination IP Address set to zero, the reply will contain the network address in both the source and destination fields of the IP header.

From the description above one can understand that the ICMP Information request and reply mechanism was intended to be used locally.

The RARP, BOOTP & DHCP protocols provide better mechanisms for hosts to discover its own IP address.

<sup>11</sup> `icmpush` was written by Stayer of [hispahack.ccc.de/](http://hispahack.ccc.de/).

# ICMP Usage in Scanning Version 2.5

A more accurate table that lists which operating systems would answer to an ICMP ECHO request aimed at their Network / Broadcast address is given below:

| Operating System                         | Echo Request<br>Broadcast |
|--|---------------------------|
| Debian GNU/ LINUX 2.2, Kernel 2.4 test 2 | +                         |
| Redhat LINUX 6.2 Kernel 2.2.14           | +                         |
| FreeBSD 4.0                              | -                         |
| FreeBSD 3.4                              | -                         |
| OpenBSD 2.7                              | -                         |
| OpenBSD 2.6                              | -                         |
| NetBSD                                   | -                         |
| Solaris 2.5.1                            | +                         |
| Solaris 2.6                              | +                         |
| Solaris 2.7                              | +                         |
| Solaris 2.8                              | +                         |
| HP-UX v10.20                             | +                         |
| AIX                                      | -                         |
| ULTRIX                                   | -                         |
| Windows 95                               | -                         |
| Windows 98                               | -                         |
| Windows 98 SE                            | -                         |
| Windows ME                               | -                         |
| Windows NT 4 WRKS SP 3                   | -                         |
| Windows NT 4 WRKS SP 6a                  | -                         |
| Windows NT 4 Server SP4                  | -                         |
| Windows 2000 Professional (and SP1)      | -                         |
| Windows 2000 Server (and SP1)            | -                         |

Table 1: Which Operating Systems would answer to an ICMP ECHO Request aimed at the Broadcast Address of the Network they reside on?

Countermeasure: Block the IP directed broadcast on the border router.

## 2.4 Non-ECHO ICMP

ICMP ECHO is not the only ICMP query message type available with the ICMP protocol:

Non-ECHO ICMP messages are being used for more advanced ICMP scanning techniques (not only probing hosts, but network devices, such as a router, as well).

The group of ICMP query message types includes the following:

ECHO Request (Type 8), and Reply (Type 0)  
Time Stamp Request (Type 13), and Reply (Type 14)  
Information Request (Type 15), and Reply (Type 16)  
Address Mask Request (Type 17), and Reply (Type 18)  
Router Solicitation (Type 10), and Router Advertisement (Type 9)

# ICMP Usage in Scanning Version 2.5

## 2.4.1 ICMP Time Stamp Request (Type 13) and Reply (Type 14)

The *ICMP Time Stamp Request and Reply* allows a node to query another for the current time. This allows a sender to determine the amount of latency that a particular network is experiencing. The sender initializes the identifier (used to identify Timestamp requests aimed at different destination hosts) and sequence number (if multiple Timestamp requests are sent to the same destination host), sets the originate time stamp and sends it to the recipient.

The receiving host fills in the receive and transmit time stamps, change the type of the message to time stamp reply and returns it to the recipient. The time stamp is the number of milliseconds elapsed since midnight UT (GMT).

The originate time stamp is the time the sender last touched the message before sending it, the receive time stamp is the time the recipient first touched it on receipt, and the Transmit time stamp is the time the receiver last touched the message on sending it.

|                     |   |      |                 |    |
|---------------------|---|------|-----------------|----|
| 0                   | 4 | 8    | 16              | 31 |
| Type                |   | Code | Checksum        |    |
| Identifier          |   |      | Sequence Number |    |
| Originate timestamp |   |      |                 |    |
| Receive timestamp   |   |      |                 |    |
| Transmit timestamp  |   |      |                 |    |

Figure 4: ICMP Time Stamp Request & Reply message format

As RFC 1122 state, a *host may* implement Timestamp and Timestamp Reply. If they are implemented a host must follow this rules:

- o Minimum variability delay in handling the Timestamp request.
- o The receiving host *must* answer to every Timestamp request that he receives.
- o An ICMP Timestamp Request to an IP Broadcast or IP Multicast address *may* be silently discarded.
- o The IP source address in an ICMP Timestamp reply *must* be the same as the specific-destination address of the corresponding Timestamp request message.
- o If a source-route option is received in a Timestamp request, the return route *must* be reserved and used as a Source Route option for the Timestamp Reply option.
- o If a Record Route and/or Timestamp option is received in a Timestamp request, this option(s) *should* be updated to include the current host and included in the IP header of the Timestamp Reply message.

Receiving an ICMP Timestamp Reply would reveal an alive host (or a networking device) that has implemented the ICMP Timestamp messages.



# ICMP Usage in Scanning Version 2.5

In the next example I have sent an ICMP Time Stamp Request, using the `icmpush`<sup>11</sup> tool, to a Redhat 6.1 LINUX, Kernel 2.2.12 machine:

```
[root@stan /root]# icmpush -tstamp 192.168.5.5
kenny.sys-security.com -> 13:48:07
```

## Snort Trace:

```
01/26-13:51:29.342647 192.168.5.1 -> 192.168.5.5
ICMP TTL:254 TOS:0x0 ID:13170
TIMESTAMP REQUEST
88 16 D8 D9 02 8B 63 3D 00 00 00 00 00 00 00 00 .....C=.....

01/26-13:51:29.342885 192.168.5.5 -> 192.168.5.1
ICMP TTL:255 TOS:0x0 ID:6096
TIMESTAMP REPLY
88 16 D8 D9 02 8B 63 3D 02 88 50 18 02 88 50 18 .....C=...P...P.
2A DE 1C 00 A0 F9 *.....
```

When I have sent an ICMP Time Stamp Request to a Windows NT WRKS 4.0 SP4 machine, I got no reply. Again, this is not an abnormal behavior from the Microsoft Windows NT machine, just an implementation choice as RFC 1122 states.

**Countermeasure:** Block ICMP Time Stamp Requests coming from the Internet on the border Router and/or Firewall.

## 2.4.2 ICMP Information Request (Type 15) and Reply (Type 16)

The *ICMP Information Request/Reply* pair was intended to support self-configuring systems such as diskless workstations at boot time, to allow them to discover their network address.

The sender fills in the request with the Destination IP address in the IP Header set to zero (meaning this network). The request may be sent with both Source IP Address and Destination IP Address set to zero. The sender initializes the identifier and the sequence number, both used to match the replies with the requests, and sends out the request. The ICMP header code field is zero.

If the request was issued with a non-zero Source IP Address the reply would only contain the network address in the Source IP Address of the reply. If the request had both the Source IP Address and the Destination IP Address set to zero, the reply will contain the network address in both the source and destination fields of the IP header.

From the description above one can understand that the ICMP Information request and reply mechanism was intended to be used locally.

The RARP, BOOTP & DHCP protocols provide better mechanisms for hosts to discover its own IP address.

<sup>11</sup> `icmpush` was written by Slayer of hispahack <http://hispahack.ccc.de/>.

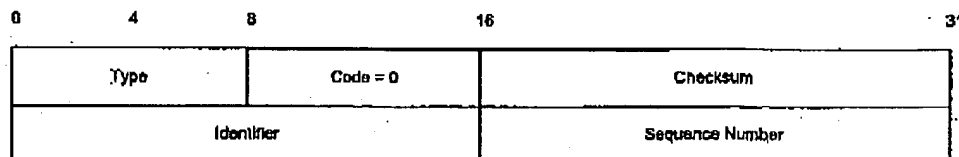
ICMP Usage in Scanning  
Version 2.5

Figure 5: ICMP Information Request &amp; Reply message format

The Information Request & Reply mechanism is now obsolete as stated in RFC 1122, and RFC 1812<sup>12</sup>. A router should not originate or respond to these messages; A host should not implement these messages.

Demands on one hand and reality on the other.

RFC 792 specifies that the Destination IP address should be set to zero, this mean that hosts that do not reside on the same network cannot send these ICMP query type.

But what would happen if we would send an ICMP Information Request with the Destination IP address set to a specific IP address of a host out in the void?

The next example illustrates that some operating systems would answer these queries even if not issued from the same network. The ICMP Information Request queries we are sending are not really RFC compliant because of the difference in the Destination IP address.

Those operating systems that answer our queries work in contrast to the RFC guidelines as well. We would see in the next example why.

In the next example I have sent an ICMP Information Request, using the SING<sup>13</sup> tool, to an AIX machine:

```
[root@aik icmp]# ./sing -info host_address14
SINGing to host_address (ip_address): 8 data bytes
8 bytes from ip_address: icmp_seq=0 ttl=238 Info Reply
8 bytes from ip_address: icmp_seq=1 ttl=238 Info Reply
8 bytes from ip_address: icmp_seq=2 ttl=238 Info Reply
8 bytes from ip_address: icmp_seq=3 ttl=238 Info Reply

--- host_address sing statistics ---
5 packets transmitted, 4 packets received, 20% packet loss
```

The tcpdump trace:

```
19:56:37.943679 ppp0 > x.x.x.x > y.y.y.y: icmp: information request
4500 001c 3372 0000 ff01 18a7 xxxx xxxx
yyyy yyyy 0f00 bee3 321c 0000
19:56:38.461427 ppp0 < y.y.y.y > x.x.x.x: icmp: information reply
4500 001c 661b 0000 ee01 f6fd yyyy yyyy
```

<sup>12</sup> RFC 1812: Requirements for IP Version 4 Routers, <http://www.ietf.org/rfc/rfc1812.txt>. As the RFC states this mechanism is now obsolete - A router should not originate or respond to these messages; A host should not implement these messages.

<sup>13</sup> SING written by Alfredo Andrés Omella, can be found at <http://sourceforge.net/projects/sing>.

<sup>14</sup> Since I have queried a production system for this test, with a permission of the owners, I do not wish to identify it.

ICMP Usage in Scanning  
Version 2.5

xxxx xxxx 1000 bde3 321c 0000

Lets do a quick analysis of the trace.

The ICMP Information Request:

| Value       | Field                                | Additional Information |
|-------------|--------------------------------------|------------------------|
| 4           | 4-Bit Version                        | IP Version 4           |
| 5           | 4-Bit Header Length                  | 4 x DWORD = 20 Bytes   |
| 00          | 8-Bit TOS                            | TOS=0                  |
| 00 1c       | 16-Bit Total Length                  |                        |
| 33 72       | 16-Bit Identification                |                        |
| 00 00       | 3-Bit Flags + 13-bit Fragment Offset |                        |
| ff          | 8-Bit TTL                            | TTL=255                |
| 01          | 8-Bit Protocol                       | 1=ICMP                 |
| 18 a7       | 16-Bit Header Checksum               |                        |
| 8b 5c d0 15 | 32-bit Source IP Address             | 139.92.208.21          |
| xx xx xx xx | 32-Bit Destination IP Address        |                        |
| 0f          | 8-Bit Type                           | Type=15                |
| 00          | 8-Bit Code                           | Code=0                 |
| ba e3       | 16-Bit Checksum                      |                        |
| 32 1c       | 16-Bit Identifier                    |                        |
| 00 00       | 16-Bit Sequence Number               |                        |

The ICMP Information Reply:

| Value       | Field                                | Additional Information |
|-------------|--------------------------------------|------------------------|
| 4           | 4-Bit Version                        | IP Version 4           |
| 5           | 4-Bit Header Length                  | 4 x DWORD = 20 Bytes   |
| 00          | 8-Bit TOS                            | TOS=0                  |
| 00 1c       | 16-Bit Total Length                  |                        |
| 66 1b       | 16-Bit Identification                |                        |
| 00 00       | 3-Bit Flags + 13-bit Fragment Offset |                        |
| ee          | 8-Bit TTL                            | TTL=238                |
| 01          | 8-Bit Protocol                       | 1=ICMP                 |
| F6 fd       | 16-Bit Header Checksum               |                        |
| xx xx xx xx | 32-bit Source IP Address             |                        |
| 8b 5c d0 15 | 32-Bit Destination IP Address        | 139.92.208.21          |
| 10          | 8-Bit Type                           | Type=16                |
| 00          | 8-Bit Code                           | Code=0                 |
| bd e3       | 16-Bit Checksum                      |                        |
| 32 1c       | 16-Bit Identifier                    |                        |
| 00 00       | 16-Bit Sequence Number               |                        |

Instead of having the network address in the Source IP Address we are getting the IP address of the host.

Does the reply compliant with RFC 792 regarding this issue? Basically yes, because the RFC does not specify an accurate behavior.

The RFC states: "To form a information reply message, the source and destination addresses are simply reversed, the type code changes to 16, and the checksum recomputed".

#### ICMP Usage in Scanning Version 2.5

This means that if the ICMP Information Request is coming from outside (Destination is not zero) of the network in question, the network address would not be revealed. But still a host could be revealed if he answers the request.

The request is not compliant with the RFC in my opinion because it does not fulfill its job -- getting the network address.

**Countermeasure:** Block ICMP Information Requests coming from the Internet on the border Router and/or Firewall.

#### 2.4.3 ICMP Address Mask Request (Type 17) and Reply (Type 18)

The *ICMP Address Mask Request* (and Reply) is intended for diskless systems to obtain its subnet mask in use on the local network at bootstrap time. Address Mask request is also used when a node wants to know the address mask of an interface. The reply (if any) contains the mask of that interface.

Once a host has obtained an IP address, it could then send an Address Mask request message to the broadcast address of the network they reside on (255.255.255.255). Any host on the network that has been configured to send address mask replies will fill in the subnet mask, change the type of the message to address mask reply and return it to the sender.

RFC 1122 states that the Address Mask request & reply query messages are entirely optional.

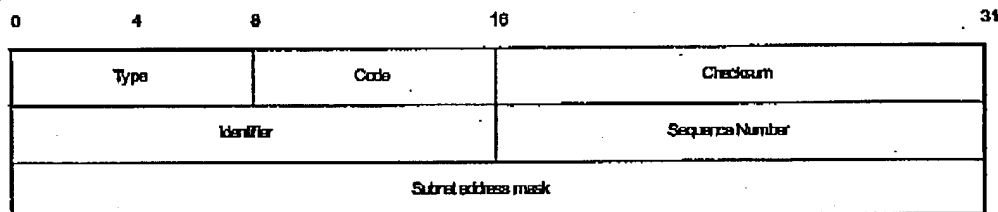


Figure 6: ICMP Address Mask Request & Reply message format

RFC 1122 also states that a system that has implemented ICMP Address Mask messages *must* not send an Address Mask Reply unless it is an authoritative agent for address masks.

Usually an Address Mask request would be answered by a gateway.

Receiving an Address Mask Reply from a host would reveal an alive host that is an authoritative agent for address masks. It will also allow a malicious computer attacker to gain knowledge about your network's configuration. This information can assist the malicious computer attacker in determining your internal network structure, as well as the routing scheme.

Please note that a Router *must* implement ICMP Address Mask messages. This will help identify routers along the path to the targeted network (It can also reveal internal routers if this kind of traffic is allowed to reach them).

If the Router is following RFC 1812 closely, it should not forward on an Address Mask Request to another network.

# ICMP Usage in Scanning Version 2.5

ICMP Address Mask Request aimed at a LINUX machine would not trigger an ICMP Address Mask Reply, nor a request aimed at a Microsoft Windows NT 4 Workstation SP 6a box.

In the next example I have sent an ICMP Address Mask Request to the broadcast address (192.168.5.255) of a class C network 192.168.5.0, spoofing the source IP to be 192.168.5.3:

```
[root@stan /root]# icmpush -vv -mask -sp 192.168.5.3 192.168.5.255
-> ICMP total size = 12 bytes
-> Outgoing interface = 192.168.5.1
-> MTU = 1500 bytes
-> Total packet size (ICMP + IP) = 32 bytes
ICMP Address Mask Request packet sent to 192.168.5.255 (192.168.5.255)
```

Receiving ICMP replies ...

```
-----
192.168.5.3 ...
  Type = Address Mask Request (0x11)
  Code = 0x0      Checksum = 0xBF87
  Id = 0x3B7      Seq# = 0x3CB0
-----
```

icmpush: Program finished OK

The snort trace:

```
--> Snort! <*-
Version 1.5
By Martin Roesch (roesch@clark.net, www.clark.net/~roesch)
Kernel filter, protocol ALL, raw packet socket
Decoding Ethernet on interface eth0
02/15-13:47:37.179276 192.168.5.3 -> 192.168.5.255
ICMP TTL:254 TOS:0x0 ID:13170
ADDRESS REQUEST
B9 03 8E 49 00 00 00 00      ...I....
```

No answer was received from the LINUX machines or from the Microsoft Windows NT Workstation 4 SP 6a machine on our test lab.

When I have tried to map which operating systems would answer (if at all) the ICMP Address Mask Requests, I have discovered that SUN Solaris is very cooperative with this kind of query<sup>15</sup>:

```
[root@aik icmp]# ./sing -mask -c 1 IP_Address16
SINGing to IP_Address (IP_Address): 12 data bytes
12 bytes from IP_Address: icmp_seq=0 ttl=241 mask=255.255.255.0

--- IP_Address sing statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
[root@aik icmp]#
```

The Tcpdump trace:

<sup>15</sup> The -c 1 option enable SING to send only one ICMP datagram. The parameter can be changed to any desired value.  
<sup>16</sup> The real IP Address and the Host address were replaced.

ICMP Usage In Scanning  
Version 2.5

```

20:02:07.402229_ppp0 > x.x.x.x > y.y.y.y: icmp: address mask request
                        4500 0020 3372 0000 ff01 70a7 xxxx xxxx
                        YYY YYY Y 1100 afe3 3f1c 0000 0000 0000
20:02:07.831426_ppp0 < y.y.y.y > x.x.x.x: icmp: address mask is
0xffffffff00 (DF)
                        4500 0020 3617 4000 f101 3c02 YYY YYY
                        xxxx xxxx 1200 afe2 3f1c 0000 ffff ff00

```

Our two last examples would be an ICMP Address Mask request aimed at a router (which must implement ICMP Address Mask Messages) and at a switch.

The following is an Address Mask Request sent to a Cisco Catalyst 5505 with OSS v4.5:

```

inferno:/tmp# sing -mask -c 1 10.13.58.240
SINGing to 10.13.58.240 (10.13.58.240): 12 data bytes
12 bytes from 10.13.58.240: icmp_seq=0 ttl=60 mask=255.255.255.0
--- 10.13.58.240 sing statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
inferno:/tmp#

inferno:~# tcpdump -tnxv -s 1600 icmp
tcpdump: listening on xl0
10.13.58.199 > 10.13.58.240: icmp: address mask request (ttl 255, id
13170)
0000 : 4500 0020 3372 0000 FF01 FE99 0A0D 3AC7 E.. 3r.....
0010 : 0A0D 3AF0 1100 6BF7 8308 0000 0000 0000 .....k.....

10.13.58.240 > 10.13.58.199: icmp: address mask is 0xffffffff00 (ttl 60,
id 20187)
0000 : 4500 0020 4EDB 0000 3C01 A631 0A0D 3AF0 E.. N...<..l...
0010 : 0A0D 3AC7 1200 6BF6 8308 0000 FFFF FF00 .....k.....
0020 : 0000 0000 0000 0000 0000 0000 0000 .....
^C
79 packets received by filter
0 packets dropped by kernel
inferno:~#

```

The last example is an ICMP Address Mask request sent to an Intel 8100 ISDN Router on our network:

```

[root@aik icmp]# ./sing -mask 10.0.0.254
SINGing to 10.0.0.254 (10.0.0.254): 12 data bytes
12 bytes from 10.0.0.254: icmp_seq=0 ttl=64 mask=255.255.255.0
12 bytes from 10.0.0.254: icmp_seq=1 ttl=64 mask=255.255.255.0
12 bytes from 10.0.0.254: icmp_seq=2 ttl=64 mask=255.255.255.0
--- 10.0.0.254 sing statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
[root@aik icmp]#

```

The tcpdump trace:

# ICMP Usage in Scanning Version 2.5

```
[root@aik /root]# tcpdump -x icmp
Kernel filter, protocol ALL, datagram packet socket
tcpdump: listening on all devices
16:34:30.666687 eth0 > 10.0.0.105 > 10.0.0.254: icmp: address mask
request
          4500 0020 3372 0000 ff01 7304 0a00 0069
          0a00 00fe 1100 0afd e402 0000 0000 0000
16:34:30.667961 eth0 < 10.0.0.254 > 10.0.0.105: icmp: address mask is
0xffffffff
          4500 0020 2cb7 0000 4001 38c0 0a00 00fe
          0a00 0069 1200 0afc e402 0000 ffff ff00
          0000 0000 0000 0000 0000 0000 0000
```

**Countermeasure:** Block ICMP Address Mask Requests coming from the Internet on the border Router and/or Firewall.

## 2.5 Non-ECHO ICMP Sweeps

We can query multiple hosts using a Non-ECHO ICMP query message type. This is referred as a Non-ECHO ICMP sweep.

Who would answer our query?

Hosts that answer to the following:

- o Hosts that are in a listening state.
- o Hosts running an operating system that implemented the Non-ECHO ICMP query message type that was sent.
- o Hosts that are configured to reply to the Non-ECHO ICMP query message type (few conditions here as well, for example: RFC 1122 states that a system that implemented ICMP Address Mask messages *must not* send an Address Mask Reply unless it is an authoritative agent for address masks).

Given the conditions above, which host(s) would answer our queries?

| Operating System                         | Info. Request | Time Stamp Request | Address Mask Request |
|--|---------------|--------------------|----------------------|
| Debian GNU/ LINUX 2.2, Kernel 2.4 test 2 | -             | +                  | -                    |
| Redhat LINUX 6.2 Kernel 2.2.14           | -             | +                  | -                    |
| FreeBSD 4.0                              | +             | +                  | -                    |
| FreeBSD 3.4                              | -             | +                  | -                    |
| OpenBSD                                  | -             | +                  | -                    |
| NetBSD                                   | -             | -                  | -                    |
| Solaris 2.5.1                            | -             | +                  | +                    |
| Solaris 2.6                              | -             | +                  | +                    |
| Solaris 2.7                              | -             | +                  | +                    |
| Solaris 2.8                              | -             | +                  | +                    |
| HP-UX v10.20                             | +             | +                  | -                    |
| AIX v4.x                                 | +             | +                  | -                    |

# ICMP Usage in Scanning Version 2.5

| Operating System          | Info. Request | Time Stamp Request | Address Mask Request |
|---------------------------|---------------|--------------------|----------------------|
| ULTRIX 4.2 - 4.5          | +             | +                  | +                    |
| Windows 95                | -             | -                  | +                    |
| Windows 98                | -             | +                  | +                    |
| Windows 98 SE             | -             | +                  | +                    |
| Windows ME                | -             | +                  | -                    |
| Windows NT 4 WRKS SP 3    | -             | -                  | +                    |
| Windows NT 4 WRKS SP 6a   | -             | -                  | -                    |
| Windows NT 4 Server SP 4  | -             | +                  | -                    |
| Windows 2000 Professional | -             | +                  | -                    |
| Windows 2000 Server       | -             | +                  | -                    |

| Networking Devices                  | Info. Request | Time Stamp Request | Address Mask Request |
|-------------------------------------|---------------|--------------------|----------------------|
| Cisco Catalyst 5505 with OSS v4.5   | +             | +                  | +                    |
| Cisco Catalyst 2900XL with IOS 11.2 | +             | +                  | -                    |
| Cisco 3600 with IOS 11.2            | +             | +                  | -                    |
| Cisco 7200 with IOS 11.3            | +             | +                  | -                    |
| Intel Express 8100 ISDN Router      | -             | -                  | +                    |

Table 2: non-ECHO ICMP Query of different Operating Systems and Networking Devices

**Countermeasure:** Block ICMP Information Requests, ICMP Address Mask Requests & ICMP Time Stamp Requests coming from the Internet on the border Router and/or Firewall.

## 2.6 Non-ECHO ICMP Broadcasts

We can send a Non-ECHO ICMP query message type to the broadcast address or to the network address of the targeted network.

The request would be broadcasted to all listening hosts on the targeted network.

Who would answer our query?

- o Hosts that are in a listening state
- o Hosts running an operating system that implemented the Non-ECHO ICMP query message type that was sent.
- o Hosts that are configured to reply to the Non-ECHO ICMP query message type (few conditions here as well, for example: a host may discard Non-ECHO ICMP query message type requests targeted at the broadcast address. For example an ICMP Timestamp Request to an IP Broadcast or IP Multicast address may be silently discarded).

Given the conditions above, the answering hosts would almost always be UNIX and UNIX-like machines. SUN Solaris, HP-UX, and LINUX are the only operating systems, from the group of operating systems I have tested, that would answer to an ICMP Timestamp Request aimed at the broadcast address of a network. HP-UX would answer Information Requests aimed at the broadcast address of a network. Non-would answer to an ICMP Address Mask Request aimed at the broadcast address of a network.



ICMP Usage in Scanning  
Version 2.5

| Operating System                         | Info. Request<br>Broadcast | Time Stamp Request<br>Broadcast | Address Mask Request<br>Broadcast |
|--|----------------------------|---------------------------------|-----------------------------------|
| Debian GNU/ LINUX 2.2, Kernel 2.4 test 2 | -                          | +                               | -                                 |
| Redhat LINUX 8.2 Kernel 2.2.14           | -                          | +                               | -                                 |
| FreeBSD 4.0                              | -                          | -                               | -                                 |
| FreeBSD 3.4                              | -                          | -                               | -                                 |
| OpenBSD 2.7                              | -                          | -                               | -                                 |
| OpenBSD 2.6                              | -                          | -                               | -                                 |
| NetBSD                                   | -                          | -                               | -                                 |
| Solaris 2.5.1                            | +                          | +                               | -                                 |
| Solaris 2.6                              | +                          | +                               | -                                 |
| Solaris 2.7                              | +                          | +                               | -                                 |
| Solaris 2.8                              | -                          | +                               | -                                 |
| HP-UX v10.20                             | +                          | +                               | -                                 |
| AXX 4.x                                  | -                          | -                               | -                                 |
| ULTRIX 4.2 - 4.5                         | -                          | -                               | -                                 |
| Windows 95                               | -                          | -                               | -                                 |
| Windows 98                               | -                          | -                               | -                                 |
| Windows 98 SE                            | -                          | -                               | -                                 |
| Windows ME                               | -                          | -                               | -                                 |
| Windows NT 4 WRKS SP 3                   | -                          | -                               | -                                 |
| Windows NT 4 WRKS SP 6a                  | -                          | -                               | -                                 |
| Windows NT 4 Server SP 4                 | -                          | -                               | -                                 |
| Windows 2000 Professional (& SP1)        | -                          | -                               | -                                 |
| Windows 2000 Server (& SP1)              | -                          | -                               | -                                 |

Table 3: Operating Systems, which would answer to requests, aimed at the Broadcast address

| Networking Devices                  | Info. Request<br>Broadcast | Time Stamp Request<br>Broadcast | Address Mask Request<br>Broadcast |
|-------------------------------------|----------------------------|---------------------------------|-----------------------------------|
| Cisco Catalyst 5505 with IOS v4.5   | +                          | +                               | +                                 |
| Cisco Catalyst 2900XL with IOS 11.2 | +                          | -                               | -                                 |
| Cisco 3600 with IOS 11.2            | +                          | -                               | -                                 |
| Cisco 7200 with IOS 11.3            | +                          | -                               | -                                 |
| Intel Express 8100 ISDN Router      | -                          | -                               | -                                 |

Table 4: Networking Devices, which would answer to requests, aimed at the Broadcast address

**Countermeasure:** Block the IP directed broadcast on the border router. Block ICMP Information Requests, ICMP Address Mask Requests & ICMP Time Stamp Requests coming from the Internet on the border Router and/or Firewall.

ICMP Usage In Scanning  
Version 2.5

### 3.0 Advanced Host Detection using the ICMP Protocol (using ICMP Error Messages generated from the probed machines)

The advanced host detection methods rely on the idea that we can use various methods in order to elicit an ICMP Error Message back from a probed machine and discover its existence. Some of the methods described here are:

- Mangling IP headers
  - Header Length Field
  - IP Options Field
- Using non-valid field values in the IP header
  - Using valid field values in the IP header
- Abusing Fragmentation
- The UDP Scan Host Detection method

With the first method we are using bad IP headers in the IP datagram that would generate an ICMP Parameter Problem error back from the probed machine to the source IP address of the probing datagram. The second method use non-valid field values in the IP header in order to force the probed machine to generate ICMP Destination Unreachable error message back to the malicious computer attacker. The third method discussed uses fragmentation to trigger an ICMP Fragment Reassembly Time Exceeded error message from the probed machine. The last method uses the UDP Scan method to elicit ICMP Port Unreachable error message back from a closed UDP port(s) on the probed host(s).

When using some of those methods we can determine if a filtering device is present and some can even discover the Access Control List a Filtering Device is forcing on the protected network.

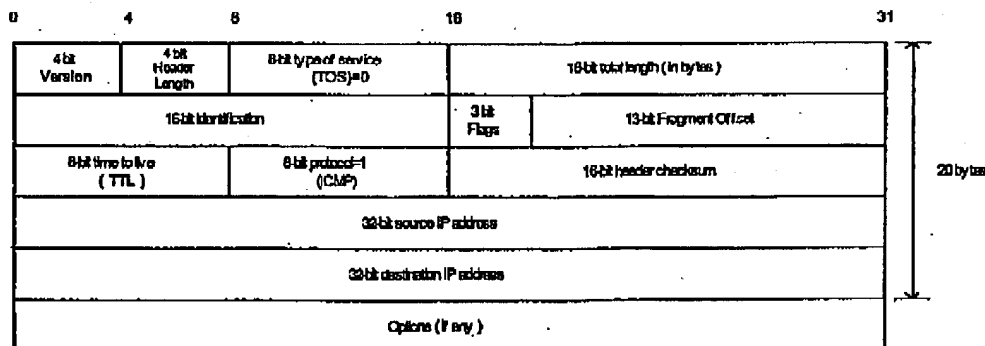


Figure 7: The IP Header

### 3.1 Sending IP Datagrams with bad IP headers fields – generating ICMP Parameter Problem error message back from probed machines

An ICMP Parameter Problem error message is sent when a router (*must* generate this message) or a host (*should* generate this message) process a datagram and finds a problem with the IP header parameters, which is not specifically covered by another ICMP error message. The ICMP Parameter Problem error message is only sent if the error caused the datagram to be discarded.

We have some variants with this type of Host Detection. We send an illegal forged datagram(s) with bad IP header field(s), that no specific ICMP error message is sent for this field(s). It will

# ICMP Usage In Scanning Version 2.5

force a Host to send back an ICMP Parameter Problem Error message with either Code 0 or Code 2 (When code 0 is used, the pointer field will point to the exact byte in the original IP Header, which caused the problem. Code 2 is sent when the Header length or the total packet length values of the IP datagram do not appear to be accurate) to the source IP address of the bad IP datagram and reveal its existence. With this type of host detection it is not relevant what would be the protocol (TCP/UDP/ICMP) embedded inside the IP datagram. All we care about is the ICMP Error messages generated by the probed machine (if any).

This method is very powerful in detecting host(s) on the probed network with direct access from the Internet, since a host should generate this error message. Routers must generate the ICMP Parameter Problem error message as well, but not all of them check the correctness of some fields inside the IP header like a host does (processing of some fields is done on the host only).

According to RFC 1122 a host should check for validity of the following fields when processing a packet<sup>17</sup>:

- Version Number – If not 4 a host must silently discard the IP packet.
- Checksum – a host should verify the IP header checksum on every received datagram and silently discard every datagram that has a bad checksum.

A router should check for the validity of the following fields when processing a packet<sup>18</sup>:

- Checksum – a router must verify the IP checksum of any packet it received, and must discard messages containing invalid checksums.

The conditions outlined eliminate the usage of this method to a limited number of fields only.

It is possible to send an IP datagram with bad field(s) in the IP header, which will get routed without getting dropped in the way to the probed machine. It should be noted that different routers perform different checks regarding the IP header (different implementation and interpretation of RFC 1812). When a router, because of a bad IP header, drops an IP packet and sends an ICMP Parameter Problem error message, it is possible to identify the manufacture of the router, and to adjust the wrong IP header field correctly according to a field, which is not checked by the manufacture of that particular router.

A router may be more forgiving than a Host regarding the IP header. This may result from the fact that a router is a vehicle for delivering the IP datagram and a Host is the Destination and the place where more processing on the datagram is done.

The downside for this method is the detection. Intrusion Detection Systems *should* alert you about abnormalities in the attacked network traffic, since not every day you receive IP packets with bad IP Header field(s).

We can use this type of Host Detection to sweep through the entire IP range of an organization and get back results, which will map all the alive hosts on the probed network with direct access from the Internet.

Even if a firewall or another filtering device is protecting the probed network we can still try to send those forged packets to an IP addresses with ports that are likely to be opened. For

<sup>17</sup> RFC 1122 – Requirements for Internet Host, <http://www.ietf.org/rfc/rfc1122.txt>

<sup>18</sup> RFC 1812 – Requirements for IPv4 Routers, <http://www.ietf.org/rfc/rfc1812.txt>

# ICMP Usage in Scanning Version 2.5

example - TCP ports 21,25,80; UDP port 53; and even try to send an ICMP message presumably coming back from a Host/Router who generated it upon receiving data from the attacked network.

In my opinion Firewalls/Filtering Devices should check the validity of those fields used to elicit the ICMP Parameter Problem error message and disallow this kind of traffic.

An example is given here using the *ISIC* tool written by Mike Frantzen<sup>19</sup>. *ISIC* sends randomly generated packets to a target computer. Its primary uses are to stress test an IP stack, to find leaks in a firewall, and to test the implementation of Intrusion Detection Systems and firewalls. The user can specify how often the packets will be fragmented; have IP options, TCP options, an urgent pointer, etc.

In the next example I have sent 20 IP Packets from a LINUX machine to a Microsoft Windows NT WRKS 4 SP4 machine. The datagrams were not fragmented nor bad IP version numbers were sent. The only weird thing sent inside the IP headers was random IP Header length, which have produced ICMP Parameter Problem Code 2 error message as I anticipated.

```
[root@stan.packetshaping]# ./isic -s 192.168.5.5 -d 192.168.5.15 -p 20
-F 0 -V 0 -I 100
Compiled against Libnet 1.0
Installing Signal Handlers.
Seeding with 2015
No Maximum traffic limiter
Bad IP Version = 0%      Odd IP Header Length = 100%
Frag'd Pcnt = 0%
```

Wrote 20 packets in 0.03s @ 637.94 pkts/s

## tcpdump trace:

```
12:11:05.843480 eth0 > kenny.sys-security.com > cartman.sys-
security.com: ip-proto-110 226 [tos 0xe6,ECT] (ttl 110, id 119,
optlen=24[ip])
12:11:05.843961 eth0 P cartman.sys-security.com > kenny.sys-
security.com: icmp: parameter problem - octet 21 Offending pkt:
kenny.sys-security.com > cartman.sys-security.com: ip-proto-110 226
[tos 0xe6,ECT] (ttl 110, id 119, optlen=24[ip]) (ttl 128, id 37776)
```

## Other fields we can use inside the IP Header

In the last example we have used a bad Header Length field value to generate an ICMP Parameter Problem code 2-error message.

An ICMP Parameter Problem would almost always result from an incorrect usage of the *IP option* field as well.

## 3.1.1 ACL Detection using IP Datagrams with bad IP headers fields

If we probe the entire IP range of the targeted network with all combinations of protocols and ports, it would draw us the targeted network topology map, and will allow us to determine the access list (ACL) a Filtering Device (if present, and not blocking outgoing ICMP Parameter Problem Error messages) is forcing.

<sup>19</sup> <http://expert.cc.purdue.edu/~frantzen/>

ICMP Usage in Scanning  
Version 2.5

This, if the filtering device does not check the validity of the mangled IP header fields, and allows the specified traffic.

**3.1.1.1 How we determine the ACL (ICMP Protocol embedded inside)?**

When the embedded protocol is ICMP, we send various ICMP message types encapsulated inside IP datagrams with bad IP header(s). If we receive a reply from a Destination IP address we have a host that is alive and an ACL, which allows this type of message of ICMP to get to the host who generated the ICMP error message (and the Parameter Problem ICMP error message is allowed from the destination host to the Internet).

If we are not getting any reply than one of three possibilities:

- The Filtering Device disallows datagrams with the kind of bad field we are using.
- The Filtering Device is filtering the type of the ICMP message we are using.
- The Filtering Device blocks ICMP Parameter Problem error messages initiated from the protected network destined to the Internet.

**3.1.1.2 How we determine the ACL (TCP or UDP Protocol embedded inside)?**

We can probe for every combination of protocol and port values inside an IP packet with bad IP header(s). If we would receive an answer it would indicate that the protocol and port we used are allowed to the probed host from the Internet, and the ICMP Parameter Problem error message is allowed from the destination host in the protected network out to the Internet. It would also indicate that the filtering device used on the targeted network is not validating the correctness of the fields we have used in order to elicit the ICMP Parameter Problem error message.

If the embedded protocol were either TCP or UDP, a reply would not be generated if:

- The Filtering Device disallows packets with the kind of bad field we are using.
- The Filtering Device filters the Protocol used.
- The Filtering Device is filtering the specific port we are using for the probe.
- The Filtering Device blocks ICMP Parameter Problem error messages initiated from the protected network destined to the Internet. In our case, the filtering device may be blocking the specific host we are probing for outgoing ICMP Parameter Problem datagrams.

Note: If we are using the IP Header Length field in order to elicit ICMP Parameter Problem error message back from the probed host(s) than the host processing the datagram may not be able to access the Protocol Information embedded inside. The reason would be the faulty calculation that would be made - where the header ends and the data portion begins.

**Countermeasure:** Block outgoing ICMP Parameter Problem from the protected network to the Internet on the Firewall & on the border Router.

Check with the manufacture of your filtering device which fields it validates on the IP header when processing a datagram.

ICMP Usage in Scanning  
Version 2.5

### 3.2 IP Datagrams with non-valid field values

This Host Detection method is based on different IP header fields within the crafted IP datagram that would have non-valid field values, which would trigger an ICMP Destination Unreachable Error message back from the probed machines.

Note that some hosts (AIX, HP-UX, Digital UNIX) may not send ICMP Protocol Unreachable messages.

#### 3.2.1 The Protocol Field example

##### 3.2.1.1 Using non-Valid (not used) IP protocol values

One such field within the IP header is the protocol field. If we will put a value, which does not represent a valid protocol number, the probed machine would elicit an ICMP Destination Unreachable - Protocol Unreachable error message back to the probed machine.

By sending this kind of crafted packets to all IP addresses within the IP address range of the probed network we can map the hosts that are directly connected to the Internet (assuming that no filtering device is present, or filtering the specific traffic).

##### 3.2.1.1.1 Detecting if a Filtering Device is present

A packet sent with a protocol value, which does not represent a valid protocol number, should elicit an ICMP Destination Unreachable - Protocol Unreachable from the probed machine. Since this value is not used (and not valid) all hosts probed, unless filtered or are AIX, HP-UX, Digital UNIX machines, should send this reply. If a reply is not received we can assume that a filtering device prevents our packet from reaching our destination or from the reply to reach us back.

##### 3.2.1.2 Using all combination of the IP protocol filed values

The difference with this variant is that we use all of the combinations available for the IP protocol field - since the IP protocol field has only 8 bits in length, there could be 256 combinations available.

NMAP 2.54 Beta 1 has integrated this variant and Fyodor have named it - IP Protocol scan. NMAP sends raw IP packets *without any further protocol header* (no payload) to each specified protocol on the target machine. If an ICMP Protocol Unreachable error message is received, the protocol is not in use. Otherwise it is assumed it is opened (or a filtering device is dropping our packets).

If our goal was Host Detection only, than using the NMAP implementation would be just fine. But if we wish to use this scan type for other purposes, such as ACL detection, than we would need the payload data as well (the embedded protocol's data).

We can determine if a filtering device is present quite easily using this scan method. If a large number of protocols (non valid values could be among those) seems to be "opened"/used (not receiving any reply - ICMP Protocol Unreachable) than we can assume a filtering device is blocking our probes (If using a packet with the protocol headers as well). If the filtering device is blocking the ICMP Protocol Unreachable error messages initiated from the protected network towards the Internet than nearly all of the 256 possible protocol values would be seemed "opened"/used.

With the current implementation with NMAP the 256 possible protocol values should be "opened" when a scan is performed against a machine inside a protected network, because a packet filter firewall (or other kind of firewall) *should* block the probe since it lacks information to validate the traffic against its rule base (information in the protocol headers such as ports for example).

# ICMP Usage in Scanning Version 2.5

In the next example I have used NMAP 2.54 Beta 1 in order to scan a Microsoft Windows 2000 Professional machine:

```
[root@catman /root]# nmap -vv -sO 192.168.1.1
```

```
Starting nmap V. 2.54BETA1 by fyodor@insecure.org (
www.insecure.org/nmap/ )
```

```
Host (192.168.1.1) appears to be up ... good.
Initiating FIN, NULL, UDP, or Xmas stealth scan against (192.168.1.1)
The UDP or stealth FIN/NULL/XMAS scan took 4 seconds to scan 254 ports.
Interesting protocols on (192.168.1.1):
(The 250 protocols scanned but not shown below are in state: closed)
```

| Protocol | State | Name |
|----------|-------|------|
| 1        | open  | icmp |
| 2        | open  | igmp |
| 6        | open  | tcp  |
| 17       | open  | udp  |

```
Nmap run completed -- 1 IP address (1 host up) scanned in 4 seconds
```

A tcpdump trace of some of the communication exchanged:

```
17:44:45.651855 eth0 > localhost.localdomain > 192.168.1.1: ip-proto-50
0 (ttl 38, id 29363)
17:44:45.652169 eth0 < 192.168.1.1 > localhost.localdomain: icmp:
192.168.1.1 protocol 50 unreachable Offending pkt:
localhost.localdomain > 192.168.1.1: ip-proto-50 0 (ttl 38, id 29363)
(ttl 128, id 578)
17:44:45.652431 eth0 > localhost.localdomain > 192.168.1.1: ip-proto-
133 0 (ttl 38, id 18)
17:44:45.652538 eth0 > localhost.localdomain > 192.168.1.1: ip-proto-
253 0 (ttl 38, id 36169)
17:44:45.652626 eth0 > localhost.localdomain > 192.168.1.1: ip-proto-92
0 (ttl 38, id 26465)
17:44:45.652727 eth0 < 192.168.1.1 > localhost.localdomain: icmp:
192.168.1.1 protocol 133 unreachable Offending pkt:
localhost.localdomain > 192.168.1.1: ip-proto-133 0 (ttl 38, id 18)
(ttl 128, id 579)
17:44:45.652760 eth0 > localhost.localdomain > 192.168.1.1: ip-proto-
143 0 (ttl 38, id 14467)
17:44:45.652899 eth0 > localhost.localdomain > 192.168.1.1: ip-proto-30
0 (ttl 38, id 30441)
17:44:45.652932 eth0 < 192.168.1.1 > localhost.localdomain: icmp:
192.168.1.1 protocol 253 unreachable Offending pkt:
localhost.localdomain > 192.168.1.1: ip-proto-253 0 (ttl 38, id 36169)
(ttl 128, id 580)
```

## 3.2.2 ACL Detection using the Protocol field

First we need to determine if a filtering device is present using a non-valid (not used) protocol number probe. If a filtering device exists then no answer (ICMP Protocol Unreachable) will be received from the probed machine, assuming it is not AIX, HP-UX or Digital UNIX<sup>20</sup>.

<sup>20</sup> You can determine this using OS finger printing methods.

# ICMP Usage in Scanning Version 2.5

If a certain protocol were not allowed through the filtering device we would not receive any ICMP error message from the probed machine. Probing for all combinations of protocols and ports against an IP range of a targeted network using non-valid and valid protocol values can determine the ACL a filtering device is forcing on the protected network, along with the topology map of a targeted network (hosts reachable from the Internet).

A reply would not be generated if:

- The Filtering Device filters the Protocol we are using
- The Filtering Device is filtering the specific port we are using for the probe.
- The Filtering Device blocks ICMP Destination Unreachable - Protocol Unreachable error messages initiated from the protected network destined to the Internet. In our case, the filtering device may be blocking the specific host we are probing for outgoing ICMP Destination Unreachable - Protocol Unreachable error messages.

Note: We can use this method for ACL detection but if the protocol we are using is not used on the target machine it should be blocked on the filtering device. Then, only opened TCP/UDP ports and allowed ICMP traffic could traverse the filtering device. If this kind of traffic is allowed we can have better ACL detection solutions then we outlined here.

**Countermeasure:** Block outgoing ICMP Protocol Unreachable error messages coming from the protected network to the Internet on your Firewall and/or Border Router. If you are using a firewall check that your firewall block protocols which are not supported (deny all stance).

## 3.3 Host Detection using IP fragmentation to elicit Fragment Reassembly Time Exceeded ICMP error message.

When a host receives a fragmented datagram with some of its pieces missing, and does not get the missing part(s) within a certain amount of time the host will discard the packet and generate an ICMP Fragment Reassembly Time Exceeded error message back to the sending host.

We can use this behavior as a Host Detection method, by sending fragmented datagrams with missing fragments to a probed host, and wait for an ICMP Fragment Reassembly Time Exceeded error message to be received from a live host(s), if any.

When we are using this method against all of the IP range of a probed network, we will discover the network topology of that targeted network.

In the next example I have sent a TCP fragment (with the MF bit set, using the -x option with hping2) to a Microsoft Windows ME machine:

```
[root@godfather bin]# hping2 -c 1 -x -Y y.y.y.y
ppp0 default routing interface selected (according to /proc)
HPING y.y.y.y (ppp0 y.y.y.y): NO FLAGS are set, 40 headers + 0 data
bytes
```

```
--- y.y.y.y hping statistic ---
1 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
[root@godfather bin]#
```

The topdump trace:



# ICMP Usage in Scanning Version 2.5

```
20:20:00.226064 ppp0 > x.x.x.x.1749 > y.y.y.y.0: .
1133572879:1133572879(0) win 512 (frag 31927:20@0+) (DF) (ttl 64)
4500 0028 7cb7 6000 4006 c8fd xxxx xxxx
d496 6607 06d5 0000 4390 f30f 0c13 6799
5000 0200 27a8 0000

20:21:00.033209 ppp0 < y.y.y.y > x.x.x.x: icmp: ip reassembly time
exceeded Offending pkt: [|tcp] (frag 31927:20@0+) (DF) (ttl 55) (ttl
119, id 12)
4500 0038 000c 0000 7701 6e9e yyyy yyyy
xxxx xxxx 0b01 b789 0000 0000 4500 0028
7cb7 6000 3706 d1fd xxxx xxxx yyyy yyyy
06d5 0000 4390 f30f
```

## 3.3.1 ACL Detection using IP fragmentation

This method can be used not only to map the entire topology map of the targeted network, but also to determine the ACL a firewall or a filtering device is forcing on the protected network.

Simply using all combinations of TCP and UDP with different ports, with the IP addresses from the IP range of the probed network will do it. When we receive a reply it means a host we queried is alive, the port we have used is opened on that host, and the ACL allows the protocol type and the port that was used to get to the probed machine (and the ICMP Fragment Reassembly Time Exceeded error message back from the probed machine to the Internet).

If we were not getting any reply back from the probed machine it can mean:

- The Filtering Device filters the Protocol used.
- The Filtering Device is filtering the specific port we are using for the probe.
- The Filtering Device blocks ICMP Fragment Reassembly Time Exceeded error messages initiated from the protected network destined to the Internet. In our case, the filtering device may be blocking the specific host we are probing for outgoing ICMP Parameter Problem datagrams.

### 3.3.1.1 An Example with UDP (Filtering Device Detection)

Since UDP is a stateless protocol it may be better suited for our needs here. The first datagram would be fragmented including enough UDP information in the first fragmented datagram that would be enough to verify the packet against a Firewall's Rule base. The second part of the datagram would not be sent. It would force any host that gets such a packet to send us back an ICMP Fragment Reassembly Time Exceeded error message when the time for reassembly exceeds.

If the port we were using were an open port, then the ICMP Fragment Reassembly Time Exceeded error message would be generated. If the port were closed then an ICMP Port Unreachable error message would be produced.

If a firewall is blocking our probed then *no reply* would be generated.

No reply would be an indication that traffic to the Host we probed is filtered.

### 3.3.1.2 An example with TCP

We can divide the first packet of the TCP handshake into two fragments. We would put enough TCP information in the first packet that would be enough to verify the packet against the Firewall's Rule base (this means the port numbers we are using are included in the packet). We will not

# ICMP Usage in Scanning Version 2.5

send the second part of the packet, forcing any host that gets such a packet to send us back an ICMP Fragment Reassembly Time Exceeded error message when the time for reassembly exceeds. This would indicate the host is accessible by this kind of traffic, which is allowed using the port we have specified as the destination port<sup>21</sup>.

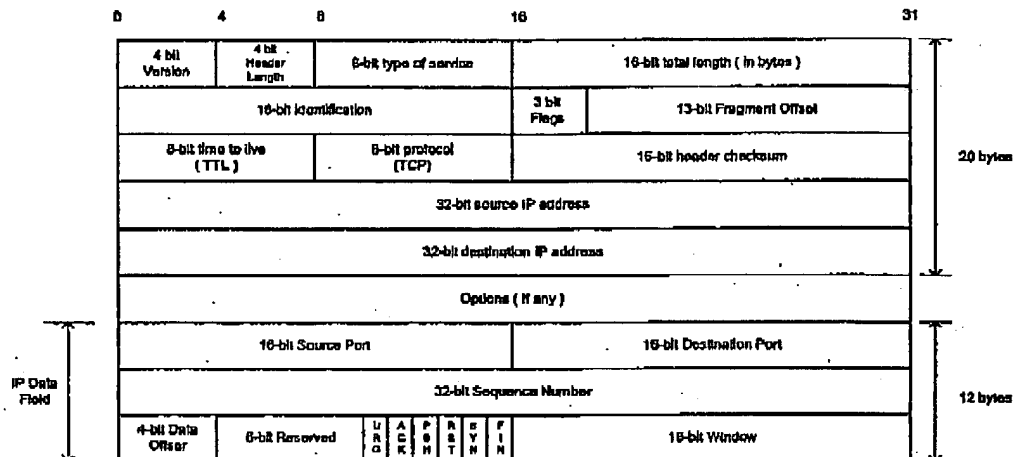


Figure 8: An Example: A TCP packet fragmented after only 12 bytes of TCP Information

If the port we are using is open, then the ICMP error message would be generated. If the port is closed then a TCP RST packet should be sent back. If a filtering device were to block our probes then no reply would be generated. No reply would be an indication that traffic to the host we probed is filtered or the filtering device requires that the first TCP packet would not be fragmented (which is a legitimate requirement).

## 3.3.1.3 An Example with ICMP

We can do the same with encapsulating the ICMP protocol. When doing so the ICMP fragmented packets should sound the sirens when an Intrusion Detection system (if deployed) sees them. There is no reason to fragment an ICMP datagram.

If we think of sending fragmented ICMP through a bad filtering device product then we should at least include the first 4 bytes of the ICMP header with the IP datagram.

**Countermeasure:** Block outgoing ICMP Fragment Reassembly Time Exceeded Error messages.

## 3.4 Host Detection using UDP Scans, or why we wait for the ICMP Port Unreachable

How can we determine if a host is alive using a UDP probe? – We use the UDP scan method that uses ICMP Port Unreachable error message that may be generated from probed hosts as indicator of alive hosts. With this method we are sending a UDP datagram with 0 bytes of data to a UDP port on the attacked machine. If we have sent the datagram to a closed UDP port we will

<sup>21</sup> In a case where a firewall is validating that the first packet is not fragmented, we can fragment another one instead. But then this scanning method would not be any different from any other scanning method using TCP flags combinations.

# ICMP Usage In Scanning Version 2.5

receive an ICMP Port Unreachable error message. If the port is opened, we would not receive any reply.

When a filtering device is blocking UDP traffic aimed at the attacked machine, it would copycat the behavior pattern as with opened UDP ports.

If we probe a large number of UDP ports on the same host and we do not receive a reply from a large number of ports, it would look like that a large number of probed UDP ports are opened. While a filtering device is probably blocking the traffic and nearly all of the ports are closed.

How can we remedy this?

We can set a threshold number of non-answering UDP ports, when reached we will assume a filtering device is blocking our probes.

Fyodor has implemented a threshold with NMAP 2.3 BETA 13, so when doing a UDP scan and not receiving an answer from a certain number of ports, it would assume a filtering device is monitoring the traffic, rather than reporting those ports as opened.

## 3.4.1 A Better Host Detection Using UDP Scan

We will take the UDP scan method and tweak it a bit for our needs. We know that a closed UDP port will generate an ICMP Port Unreachable error message indicating the state of the port - closed UDP port. We will choose a UDP port that should be definitely closed (according to the IANA list of assigned ports [ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers](http://ftp.isi.edu/in-notes/iana/assignments/port-numbers)). For example we can use port 0 (but it would reveal our probe pretty easily).

Based on the fact that sending a UDP datagram to a closed port should elicit an ICMP Port Unreachable, we would send one datagram to the port we have chosen, than:

- If no filtering device is present we will receive an ICMP Port Unreachable error message, which will indicate that the Host is alive (or if this traffic is allowed by the filtering device).
- If no answer is given - a filtering device is covering that port.

In the next example I have used the HPING2<sup>22</sup> tool to send one UDP datagram to host 192.168.5.5 port 50, which was closed:

```
[root@stan /root]# hping2 -2 192.168.5.5 -p 50 -c 1
default routing not present
HPING 192.168.5.5 (eth0 192.168.5.5): udp mode set, 28 headers + 0 data
bytes
ICMP Port Unreachable from 192.168.5.5 (kenny.sya-security.com)
```

```
--- 192.168.5.5 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

```
--> Snort! <*-
Version 1.5
By Martin Roesch (roesch@clark.net, www.clark.net/~roesch)
Kernel filter, protocol ALL, raw packet socket
```

<sup>22</sup> HPING2 written by antirez, <http://www.kyuzz.org/antirez/hping/>.

# ICMP Usage In Scanning Version 2.5

## Decoding Ethernet on interface eth0

03/12-12:54:47.274096 192.168.5.1:2420 -> 192.168.5.5:50

UDP TTL:64 TOS:0x0 ID:57254

Len: 8

03/12-12:54:47.274360 192.168.5.5 -> 192.168.5.1

ICMP TTL:255 TOS:0xC0 ID:0

DESTINATION UNREACHABLE: PORT UNREACHABLE

00 00 00 00 45 00 00 1C DF A6 00 00 40 11 0F D4 ....E.....@...

C0 A8 05 01 C0 A8 05 05 09 74 00 32 00 08 6A E1 .....t.2..j.

We can use the port number we have chosen, or a list of UDP ports that are likely not being used, and query all the IP range of an attacked network. Getting a reply back would reveal a live host. No reply would mean a filtering device is covering those hosts UDP traffic, and probably other protocols and hosts as well.

**3.5 Using Packets bigger than the PMTU of internal routers to elicit an ICMP Fragmentation Needed and Don't Fragment Bit was Set (configuration problem)**  
If internal routers have a PMTU that is smaller than the PMTU for a path going through the border router, those routers would elicit an ICMP "Fragmentation Needed and Don't Fragment Bit was Set" error message back to the initiating host if receiving a packet too big to process that has the Don't Fragment Bit set on the IP Header, discovering internal architecture of the router deployment of the attacked network.

This is in my opinion a configuration problem causing a security hazard.

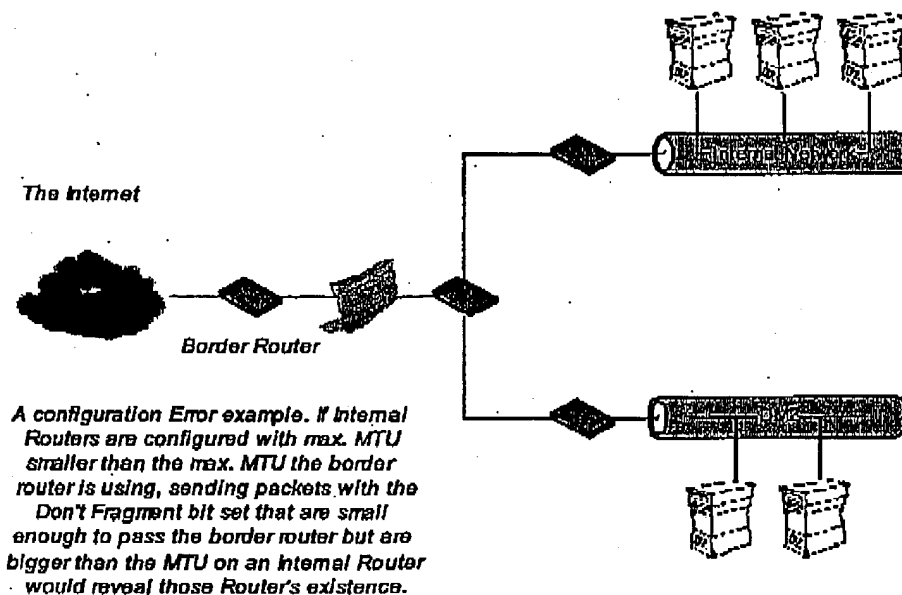


Figure 9: Using Packets bigger than the PMTU of internal routers to elicit an ICMP Fragmentation Needed and Don't Fragment Bit was Set

ICMP Usage In Scanning  
Version 2.5

#### 4.0 Inverse Mapping

Inverse Mapping is a technique used to map internal networks or hosts that are protected by a filtering devices/firewall. Usually some of those systems are not reachable from the Internet. We use routers, which will give away internal architecture information of a network, even if the question they were asked does not make any sense, for this scanning type. We compile a list of IP's that list what is not there, and use it to conclude were things probably are.

We send a number of packets to different IP's we suspect are in the IP range of the network we are probing. When a router, either an exterior or interior, gets those packets for further processing, it looks at the IP address and makes decisions of routing based on it solely. When a router gets a packets with an IP which is not used in the IP space / network segment of the part of the probed network he serves, the router will elicit an ICMP Host Unreachable (Generated by a router if a route to the destination host on a directly connected network is not available -- the destination host does not respond to ARP) or ICMP Time Exceeded error message(s) back to the originator of the datagram. If we do not get an answer about a certain IP we can assume this IP exist inside the probed network<sup>23</sup>.

#### 4.1 Inverse Mapping Using ICMP (Echo & Echo Reply)

Theoretically speaking, using any ICMP Query Message type or any ICMP Query Reply Message type in order to inverse map a network using a Router is possible.

With the next example I have sent an ICMP Echo Request to an IP that should be routed through a certain router (last hop before the host):

```
[root@cartman]# ./icmpush -vv -echo Target_IP24
-> Outgoing interface = 192.168.1.5
-> ICMP total size = 12 bytes
-> Outgoing interface = 192.168.1.5
-> MTU = 1500 bytes
-> Total packet size (ICMP + IP) = 32 bytes
ICMP Echo Request packet sent to Target_IP (Target_IP)
```

Receiving ICMP replies ...

```
-----
Routers_IP ...
      Type = Time Exceeded (0xB)
      Code = 0x0      Checksum = 0xF98F
      Id = 0x0      Seq# = 0x0
-----
```

./icmpush: Program finished OK

```
ICMP TTL:254 TOS:0x0 ID:13170
ID:12291 Seq:317 ECHO
```

```
02/13-09:16:31.724400 Routers_IP -> 192.168.1.5
ICMP TTL:57 TOS:0x0 ID:7410
```

<sup>23</sup> There is also a possibility that a filtering device is blocking our probes, or the replies.

<sup>24</sup> The real IP's of the targeted host and the Router were replaced because of legal problems that might arise when the ISP's personal that was used would understand it was one of their Routers used for this experiment.

# ICMP Usage in Scanning Version 2.5

TTL EXCEEDED

```
00:13:12 prober> 192.168.2.5: icmp: echo reply
00:13:13 router> prober: icmp: host unreachable
```

Why Using ICMP Query Replies sometimes will be more beneficial than using ICMP Query Message types?

We have more chance of getting through filtering devices, that will allow replies of certain ICMP Query message types to get back to the issuing hosts. This might allow us to "penetrate" to deeper networks with our crafted reply.

## 4.2 Inverse Mapping Using Other Protocols

The technique of inverse mapping will work with other protocols as the stimulus. It will produce the same results since the destination Host (IP) will still be unreachable. The router one hop before the targeted host could not arp the host, and will issue an ICMP Host Unreachable regardless of the underlying protocol used.

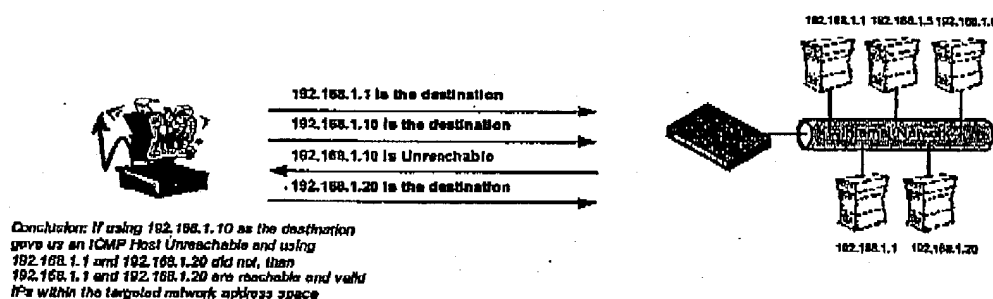


Diagram 1: The Inverse Mapping Idea

## 4.3 Patterns we might see

This type of scan will produce a number of patterns. Not always, when we will see a router issuing a host unreachable it will be because some one ment to use the inverse mapping technique.

Lets look at our first example:

```
Router_IP > The_Same_IP : icmp: host Host_A unreachable
Router_IP > The_Same_IP : icmp: host Host_D unreachable
Router_IP > The_Same_IP : icmp: host Host_G unreachable
...
Router_IP > The_Same_IP : icmp: host Host_N unreachable
...
```

The same host is being used to scan an entire IP range of a targeted network. Some of the Hosts the malicious computer attacker tries to reach are not reachable. Still, the malicious computer

# ICMP Usage in Scanning Version 2.5

attacker gets an idea about what is not reachable. Sometimes these results are the only indication that the malicious computer attacker will have about the presence of Hosts.

Lets look at the next example:

```
18:12:21.901256 Router_IP > 192.168.46.45: icmp: host x.x.x.12
unreachable
18:12:33.676136 Router_IP > 192.168.59.63: icmp: host x.x.x.12
unreachable
18:12:33.676218 Router_IP > 192.168.59.63: icmp: host x.x.x.12
unreachable
18:13:27.084221 Router_IP > 192.168.114.37: icmp: host x.x.x.12
unreachable
18:13:45.559706 Router_IP > 192.168.22.91: icmp: host x.x.x.12
unreachable
18:13:45.559856 Router_IP > 192.168.22.91: icmp: host x.x.x.12
unreachable
18:13:48.413514 Router_IP > 192.168.250.254: icmp: host x.x.x.12
unreachable
18:13:48.413681 Router_IP > 192.168.250.254: icmp: host x.x.x.12
unreachable
18:14:31.313495 Router_IP > 192.168.247.186: icmp: host x.x.x.12
unreachable
18:14:31.313624 Router_IP > 192.168.247.186: icmp: host x.x.x.12
unreachable
18:15:32.884187 Router_IP > 192.168.12.213: icmp: host x.x.x.12
unreachable
...
```

What we see here is different Hosts (changed to 192.168.x.x) failing to reach the x.x.x.12 IP. The Router is sending them all an ICMP Host Unreachable error message.

How come different Hosts (IPs) are seeking this host on such a short notice?

Probably what we are seeing is a decoy scan. A decoy scan is a type of scan, which involves multiple IPs, which are fed to the network-scanning tool as decoys. The real IP of the malicious computer attacker (or a machine he compromised) will be among those. For the defending side a question will be asked: What IP among all IPs, which are knocking on the door, is the IP the attacker was using?

With our example the IP is reported, to all seeking hosts, to be unreachable. The Router is trying to deliver the packet and fails with his ARP requests.

### ICMP Usage in Scanning Version 2.5

With this example the malicious computer (attacker) has a way to get the answers the targeted network is producing. Attacking machine on the Upstream from the target network

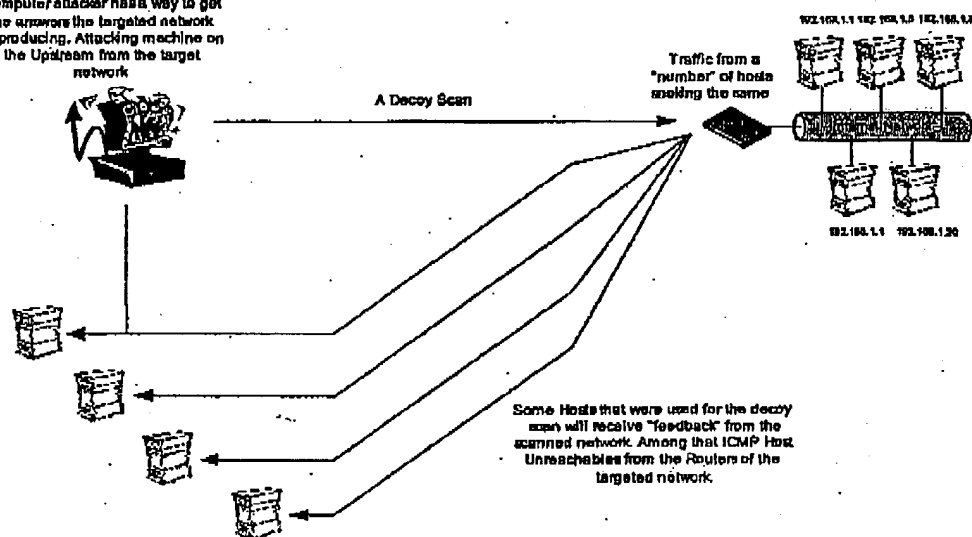


Diagram 2: A Decoy Scan Example



# ICMP Usage in Scanning Version 2.5

## 5.0 Using traceroute to Map a Network Topology

Traceroute is a Network debugging utility, which attempts to map all the hosts on a route to a certain destination host/machine.

The program sends UDP (by default) or ICMP ECHO Request<sup>25</sup> datagrams in sets of three, to a certain destination host. The first three datagram's to be sent have a Time-to-Live field value in the IP Header equals to one. The program lies on the fact that a router should decrement the TTL field value just before forwarding the datagram to another router/gateway.

If a router discovers that the Time-To-Live field value in an IP header of a datagram he process equals zero (or less) he would discard the datagram and generate an ICMP Time Exceeded: Code 0 - transit TTL expired error message back to the originating host.

This is when a successful round is completed and another set of three datagrams is sent, this time with a Time-to-Live field value greater by one than the last set.

The originating host would know at which router the datagram expired since it receives this information with the ICMP Time Exceeded in Transit error message (Source IP address of the ICMP error message would be the IP address of the router/gateway; inside the IP header + 64 bits of original data of the datagram field we would have additional information that would bound this ICMP error message to our issued traceroute command).

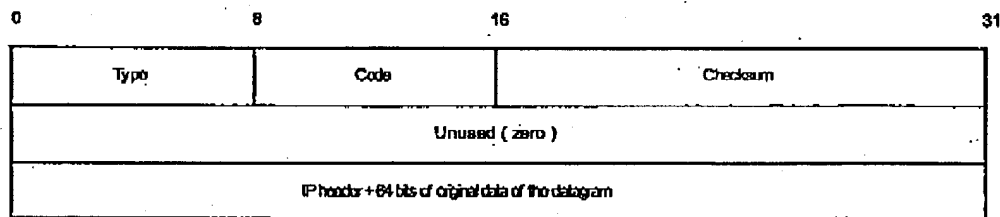


Figure 10: ICMP Time Exceeded message format

Since we increment the TTL field starting from one for each successful round (again - a round is finished when the ICMP Time Exceeded in Transit error message is received) until we receive an ICMP Port Unreachable error message (or ICMP ECHO Reply if we are using the ICMP ECHO request datagrams) from the destined machine, we map every router/gateway/host along the path to our destination.

By default, when sending UDP packets we use a destination port which is probably not used by the destination host so the UDP datagram would not be processes and an ICMP Port Unreachable error message would be generated from the destined machine. The destination port would be incremented with each probe sent.

We get ICMP responses provided there is no prohibitive filtering or any packet loss.

<sup>25</sup> Microsoft Windows NT and Microsoft Windows 2000 are using the tracert command, which use ICMP ECHO Request datagrams as its default.

# ICMP Usage in Scanning Version 2.5

The output we see is a line showing the Time-To-Live, the address of the gateway, and the round trip time of each probe. If we do not get a response back within 5 seconds an "\*" is printed, which represents no answer.

A regular traceroute example with ICMP would be<sup>26</sup>:

```
zuul:~>traceroute -I 10.0.0.10
traceroute to 10.0.0.10 (10.0.0.10), 30 hops max, 40 byte
packets
 1 10.0.0.1 (10.0.0.1) 0.540 ms 0.394 ms 0.397 ms
 2 10.0.0.2 (10.0.0.2) 2.455 ms 2.479 ms 2.512 ms
 3 10.0.0.3 (10.0.0.3) 4.812 ms 4.780 ms 4.747 ms
 4 10.0.0.4 (10.0.0.4) 5.010 ms 4.903 ms 4.980 ms
 5 10.0.0.5 (10.0.0.5) 5.520 ms 5.809 ms 6.061 ms
 6 10.0.0.6 (10.0.0.6) 9.584 ms 21.754 ms 20.530 ms
 7 10.0.0.7 (10.0.0.7) 89.889 ms 79.719 ms 85.918 ms
 8 10.0.0.8 (10.0.0.8) 92.605 ms 80.361 ms 94.336 ms
 9 10.0.0.9 (10.0.0.9) 94.127 ms 81.764 ms 96.476 ms
10 10.0.0.10 (10.0.0.10) 96.012 ms 98.224 ms 99.312 ms
```

Lets assume that a network is protected by a firewall, which blocks all incoming traffic except for traffic aimed at the DNS Machine's UDP port 53. If we would perform a regular traceroute aimed for the DNS machine's IP address, our UDP datagrams would be sent with a destination port, which is probably not used on the targeted machine, and probably blocked by a Firewall or another filtering device. The traces would stop at the firewall at the entrance point to the probed network.

```
zuul:~>traceroute 10.0.0.10
traceroute to 10.0.0.10 (10.0.0.10), 30 hops max, 40 byte
packets
 1 10.0.0.1 (10.0.0.1) 0.540 ms 0.394 ms 0.397 ms
 2 10.0.0.2 (10.0.0.2) 2.455 ms 2.479 ms 2.512 ms
 3 10.0.0.3 (10.0.0.3) 4.812 ms 4.780 ms 4.747 ms
 4 10.0.0.4 (10.0.0.4) 5.010 ms 4.903 ms 4.980 ms
 5 10.0.0.5 (10.0.0.5) 5.520 ms 5.809 ms 6.061 ms
 6 10.0.0.6 (10.0.0.6) 9.584 ms 21.754 ms 20.530 ms
 7 10.0.0.7 (10.0.0.7) 89.889 ms 79.719 ms 85.918 ms
 8 10.0.0.8 (10.0.0.8) 92.605 ms 80.361 ms 94.336 ms
 9 * * *
10 * * *
```

We need to set the port number to 53 in order to reach the DNS server. Since the traceroute program increases the port number every time it sends a UDP datagram, we need to calculate the port number to start with, so when a datagram would be processed by the Firewall<sup>27</sup> and would be examined, it would have the appropriate port and other information needed to fit with the Access Control List. If we use a simple equation we can calculate the starting port:

$$(\text{Target port} - (\text{number of hops} * \text{number of probes})) - 1$$

The number of hops (gateways) from our probing machine to the firewall is taken from our earlier traceroute. We use three probes for every query with the same TTL value, each one of them uses a different destination port number.

<sup>26</sup> All examples taken from "A Traceroute-Like Analysis of IP Packet Responses to Determine Gateway Access Control Lists" by David Goldsmith and Michael Shiffman. No real examples were provided because of legal issues.

<sup>27</sup> A firewall should not elicit any reply for any traffic destined directly for him.

# ICMP Usage in Scanning Version 2.5

```

zuul:~>tracert -p28 10.0.0.10
tracert to 10.0.0.10 (10.0.0.10), 30 hops max, 40 byte packets
 1 10.0.0.1 (10.0.0.1) 0.501 ms 0.399 ms 0.395 ms
 2 10.0.0.2 (10.0.0.2) 2.433 ms 2.940 ms 2.481 ms
 3 10.0.0.3 (10.0.0.3) 4.790 ms 4.830 ms 4.885 ms
 4 10.0.0.4 (10.0.0.4) 5.196 ms 5.127 ms 4.733 ms
 5 10.0.0.5 (10.0.0.5) 5.650 ms 5.551 ms 6.165 ms
 6 10.0.0.6 (10.0.0.6) 7.820 ms 20.554 ms 19.525 ms
 7 10.0.0.7 (10.0.0.7) 88.552 ms 90.006 ms 93.447 ms
 8 10.0.0.8 (10.0.0.8) 92.009 ms 94.855 ms 88.122 ms
 9 10.0.0.9 (10.0.0.9) 101.163 ms * *
10 * * *

```

But with the regular traceroute program we now face another difficulty. After the datagram have passed the ACL of the Firewall (and we assume the firewall lets ICMP TTL Exceeded messages out) and listed the outer leg of the Firewall itself as the next hop, the next UDP datagram sent would be with a different port number - Than again it would be blocked by the firewall.

A modification to the traceroute program has been made by Michael Shiffman<sup>28</sup> in order to stop the port incrementation. One side affect from sending traceroutes with a fixed port number, which is allowed on the firewalls ACL, is the final datagram, which normally would generate an ICMP Port Unreachable message now would not be generated since the UDP port would be in a listening state on the probed machine and would not provide an answer.

```

zuul:~>tracert -S -p53 10.0.0.15
tracert to 10.0.0.15 (10.0.0.15), 30 hops max, 40 byte packets
 1 10.0.0.1 (10.0.0.1) 0.516 ms 0.396 ms 0.390 ms
 2 10.0.0.2 (10.0.0.2) 2.516 ms 2.476 ms 2.431 ms
 3 10.0.0.3 (10.0.0.3) 5.060 ms 4.848 ms 4.721 ms
 4 10.0.0.4 (10.0.0.4) 5.019 ms 4.694 ms 4.973 ms
 5 10.0.0.5 (10.0.0.5) 6.097 ms 5.856 ms 6.002 ms
 6 10.0.0.6 (10.0.0.6) 19.257 ms 9.002 ms 21.797 ms
 7 10.0.0.7 (10.0.0.7) 84.753 ms * *
 8 10.0.0.8 (10.0.0.8) 96.864 ms 98.006 ms 95.491 ms
 9 10.0.0.9 (10.0.0.9) 94.300 ms * 96.549 ms
10 10.0.0.10 (10.0.0.10) 101.257 ms 107.164 ms 103.318 ms
11 10.0.0.11 (10.0.0.11) 102.847 ms 110.158 ms *
12 10.0.0.12 (10.0.0.12) 192.196 ms 185.265 ms *
13 10.0.0.13 (10.0.0.13) 168.151 ms 183.238 ms 183.458 ms
14 10.0.0.14 (10.0.0.14) 218.972 ms 209.388 ms 195.686 ms
15 10.0.0.15 (10.0.0.15) 236.102 ms 237.208 ms 230.185 ms

```

<sup>28</sup> <http://www.packetfactory.net>

ICMP Usage in Scanning  
Version 2.5

## 6.0 The usage of ICMP in Active Operating System Fingerprinting Process

Finger Printing is the art of Operating System Detection.

A malicious computer attacker needs few pieces of information before launching an attack. First, a target, a host detected using a host detection method. The next piece of information would be the services that are running on that host. This would be done with one of the Port Scanning methods. The last piece of information would be the operating system used by the host.

The information would allow the malicious computer attacker to identify if the targeted host is vulnerable to a certain exploit aimed at a certain service version running on a certain operating system.

In this section I have outlined the ICMP methods for this type of scan. Few methods are new and were discovered during this research.

### Using Regular ICMP Query Messages

#### 6.1 The "Who answer what?" approach

The question "Which operating system answer for what kind of ICMP Query messages?" help us identify certain groups of operating systems.

For example, LINUX and \*BSD based operating systems with default configuration answer for ICMP Echo requests and for ICMP Timestamp Requests. Until Microsoft Windows 2000 family of operating systems has been released it was a unique combination for these two groups of operating systems. Since the Microsoft Windows 2000 operating system family mimics the same behavior (yes mimic), it is no longer feasible to make this particular distinction.

Microsoft might have been thinking that this way of behavior might hide Microsoft windows 2000 machines in the haze. As we will see with the examples given in this research paper they have much more to learn.

The thing is there is no clear distinction between one operating system to another based on this data. We can only group them together and try other methodologies in order to divide those groups a bit more<sup>29</sup>.

Other data we might use is "Which operating system answer for queries aimed at the broadcast / network address of the network they reside on?"

For the complete mapping of the operating systems I have queried for this research please see "Appendix C: Mapping Operating Systems for answering/ discarding ICMP query message types", and "Appendix E: ICMP Query Message Types aimed at a Broadcast Address".

Two examples are given in this text for the usage of Operating System fingerprinting with the "Who answer what?" approach.

<sup>29</sup> Note: If the PMTU Discovery process using ICMP Echo requests is enabled with HP-UX 10.30 & 11.0x operating systems then our simple query will trigger a "retaliation" from those machines, enabling us to identify them very easily. For more information on this issue see section 6.2

**ICMP Usage In Scanning  
Version 2.5****6.1.2 Using ICMP Information Requests**

Because of the fact, that only few operating systems would reply to ICMP Information requests, we can group them together.

From the Information given in table 2 in Section 2.5, we can conclude that HP-UX 10.20, AIX, ULTRIX & Open-VMS would be the only operating systems (among those I have tested) that would produce an ICMP Information reply for these queries.

We can further distinguish between those operating systems if we would send an ICMP Address Mask Request and wait for the reply from the systems in question. AIX and HP-UX operating systems would not answer the query, while the ULTRIX & Open-VMS would.

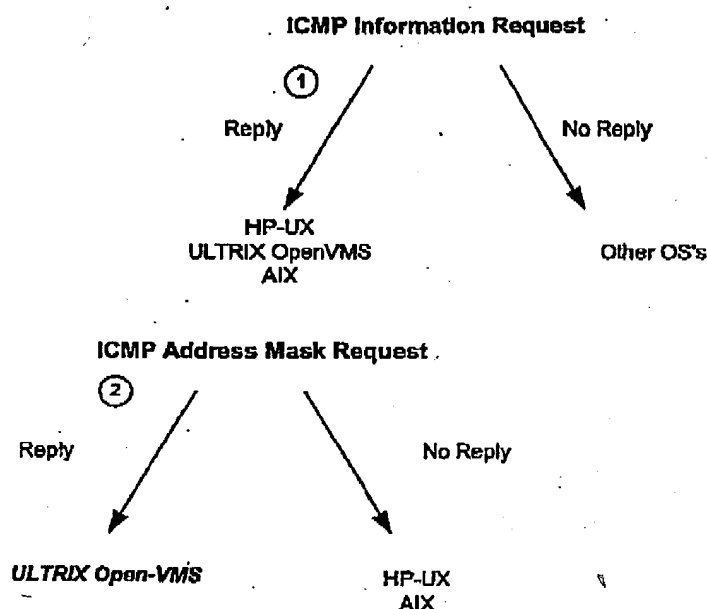


Diagram 3: Finger Printing Using ICMP Information Request Combined with ICMP Address Mask Request

**6.1.3 Identifying Operating Systems according to their replies for non-ECHO ICMP requests aimed at the broadcast address**

If IP directed broadcasts are not blocked, then we can identify the answering machines quite easily.

The first step is sending an ICMP Timestamp request aimed at the broadcast address of a targeted network. The operating systems who would answer would include SUN Solaris, HP-UX 10.20, and LINUX (Kernel version 2.2.x). We can further identify those operating systems by sending an ICMP Information request aimed at the broadcast address of the targeted network. HP-UX 10.20 would answer the query while SUN Solaris and LINUX would not. To distinguish between those two we would send an ICMP Address Mask request to the IPs that did not answer in the previous step. SUN Solaris would reply to the query while LINUX machines based on Kernel 2.2.x would not.

# ICMP Usage in Scanning Version 2.5

For complete Information see Section 2.6.

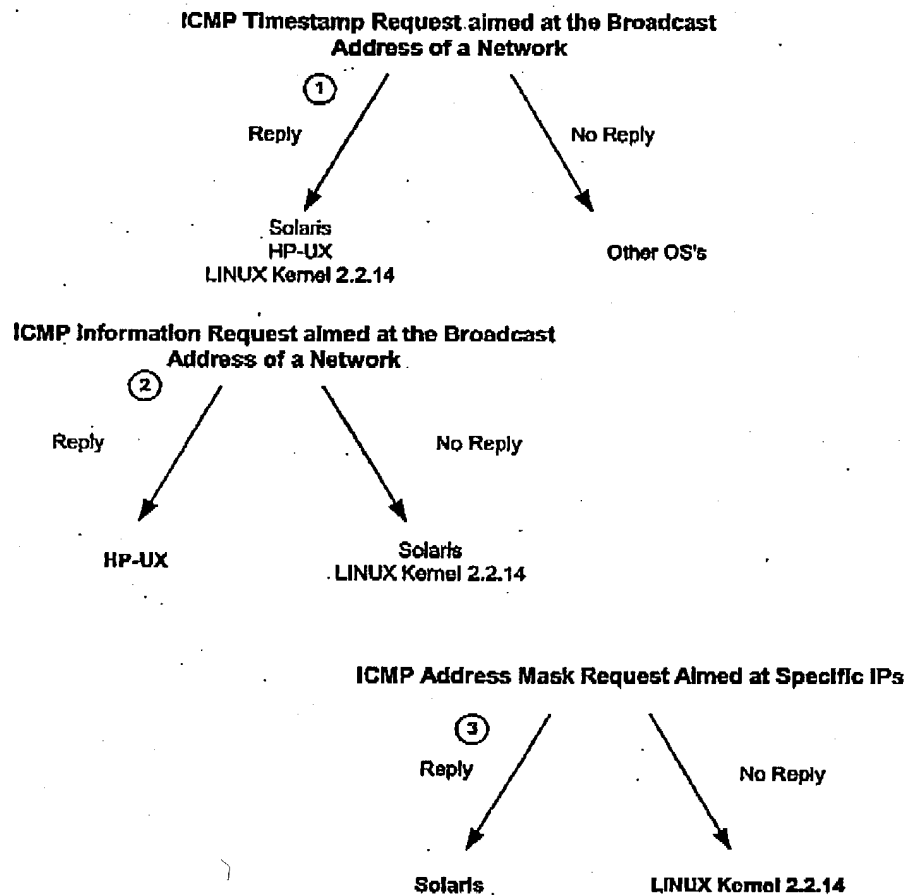


Diagram 4: Finger Printing Using non-ECHO ICMP Query Types aimed at the Broadcast Address of an Attacked Network

## 6.2 The DF Bit Playground (Identifying Sun Solaris, HP-UX 10.30, 11.0x, and AIX 4.3.x based machines)

RFC 791 defines a three bits field used for various control flags in the IP Header.

Bit 0 is the reserved flag, and must be zero.

Bit 1, is called the Don't Fragment flag, and can have two values. A value of zero (not set) is equivalent to May Fragment, and a value of one is equivalent to Don't Fragment. If this flag is set then the fragmentation of this packet at the IP level is not permitted, otherwise it is.

# ICMP Usage in Scanning Version 2.5

Bit 2, is called the More Fragments bit. It can have two values. A value of zero is equivalent to (this is the) Last Fragment, and a value of 1 is equivalent to More Fragments (are coming).

The next field in the IP header is the Fragment Offset field, which identifies the fragment location relative to the beginning of the original un-fragmented datagram (RFC 791, bottom of page 23).

A close examination of the ICMP Query replies would reveal that some operating systems would set the DF bit with their replies.

The tcpdump trace below illustrates the reply a Sun Solaris 2.7 box produced for an ICMP Echo Request:

```
17:10:19.538020 if 4 > y.y.y.y > x.x.x.x : icmp: echo request (ttl
255, id 13170)
      4500 0024 3372 0000 ff01 9602 yyyy yyyy
      xxxx xxxx 0800 54a4 8d04 0000 cbe7 bc39
      8635 0800
17:10:19.905254 if 4 < x.x.x.x > y.y.y.y : icmp: echo reply (DF) (ttl
233, id 24941)
      4500 0024 616d 4000 e901 3e07 xxxx xxxx
      yyyy yyyy 0000 5ca4 8d04 0000 cbe7 bc39
      8635 0800
```

In the recent SING CVS (12 September 2000), written by Alfredo Andres Omella, which is available from <http://sourceforge.net/projects/sing>, the option for detecting if the DF bit is set with an ICMP Query reply was added, after being request by me. The following is the same ICMP Echo request & reply, this time it is presented by SING:

```
[root@godfather bin]# ./sing -echo Host_Address
SINGing to www.openbsd.org (IP_Address): 16 data bytes
16 bytes from IP_Address: icmp_seq=0 DF! ttl=233 TOS=0 time=367.314 ms
16 bytes from IP_Address: icmp_seq=1 DF! ttl=233 TOS=0 time=320.020 ms
16 bytes from IP_Address: icmp_seq=2 DF! ttl=233 TOS=0 time=370.037 ms
16 bytes from IP_Address: icmp_seq=3 DF! ttl=233 TOS=0 time=330.025 ms

--- Host_Address sing statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 320.020/346.849/370.037 ms
```

Since [www.openbsd.org](http://www.openbsd.org) uses a Sun Solaris operating system, this matches our findings.

ICMP Query replies for an operating system maintains the same behavioral patterns. Either they set the DF bit on all ICMP query reply types or they do not.

The DF bit would be set by default with ICMP Query replies with Sun Solaris. With HP-UX 10.30, & 11.0x, and with AIX 4.3.x setting the DF Bit will vary from one queried host to another (explanation coming). It may be set with the first ICMP Query reply onwards, or after a number of ICMP Query replies. This detail will help us to distinguish between Sun Solaris, HP-UX 10.30 & 11.0x, and AIX 4.3.x operating systems.

## ICMP Usage in Scanning Version 2.5

### Why HP-UX 10.30 / 11.0 & AIX 4.3.x operating systems act this way?

HP claims to have a proprietary method in order to determine the PMTU with HP-UX v10.30, and HP-UX v11.0x using ICMP Echo requests. AIX 4.3.x do exactly the same.

The next trace will help understanding the process taken by HP-UX 10.30 & 11.0x and AIX 4.3.x.

Here I have sent an ICMP Echo request to an HP-UX 11.0 machine:

```
00:27:56.884147 ppp0 > y.y.y.y > x.x.x.x: icmp: echo request (ttl 255,
id 13170)
```

```
4500 0024 3372 0000 ff01 7c51 yyy yyy
xxxx xxxx.0800 5238 6d04 0000 dce5 c339
8b7d 0d00
```

```

00:27:57.165620 ppp0 < x.x.x.x > y.y.y.y : icmp: echo reply (ttl 236,
id 41986)

```

```

4500 0024 a402 0000 ec01 1ec1 xxxxx xxxxx
yyyy yyyy 0000 5a38 6d04 0000 dce5 c339
Bb7d 0d00

```

The first pair of ICMP Echo request and ICMP Echo reply was pretty usual. My LINUX machine sent an ICMP Echo request and received an ICMP Echo reply from the probed machine. One notable detail – the DF bit is not set in the ICMP Echo reply.

**Than something that was not expectable has happened:**

```
00:27:57.425620 ppp0 < x.x.x.x > y.y.y.y : icmp: echo request (DF) (ttl
236. id 41985)
```

[illegible]



## ICMP Usage in Scanning Version 2.5

[illegible]

## ICMP Usage in Scanning

### Version 2.5

[illegible]

```
00:27:57.435672 ppp0 > y.y.y.y > x.x.x.x: icmp: echo reply (ttl 255, id 53)
```

[illegible]

## ICMP Usage in Scanning Version 2.5

[illegible]

The HP-UX 11.x machine I have queried pinged me back! The ICMP Echo request size was 1500 bytes. It was the maximum transfer unit my Internet Connection was allowed to process. The request was sent with the DF bit set. Any router along the way, trying to fragment the request because the MTU of the destined network was smaller than the datagram's size would fail and send an ICMP Error message back stating a fragmentation was required but the don't fragment

# ICMP Usage in Scanning Version 2.5

bit was set. It would allow the sending machine to send a smaller sized datagram according to its PMTU discovery process/algorithm with ICMP. If for this ICMP Echo request an ICMP Echo reply would be received, than the PMTU is discovered.

```
00:27:57.885662 ppp0 > y.y.y.y > x.x.x.x : icmp: echo request (ttl 255, id 13170)
```

```
4500 0024 3372 0000 ff01 7c51 YYY YYY
xxxx xxxx 0800 5832 6d04 0100 dde5 c339
8383 0d00
```

```
00:27:58.155627 ppp0 < x.x.x.x > y.y.y.y : icmp: echo reply (DF) (ttl 236, id 41987)
```

```
4500 0024 a403 4000 ec01 debf xxxx xxxx
YYYY YYYY 0000 6032 6d04 0100 dde5 c339
8383 0d00
```

The following ICMP Echo Request sent from my machine to the queried HP-UX 11.x just milliseconds after my reply to the HP-UX's query was sent. It has resulted in an ICMP Echo reply coming back from the queried machine. This time the DF bit was set with the ICMP Echo reply. Rather than sending an ICMP datagram that will be fragmented somewhere along the way to the destination machine, it is more beneficial from performance perspective, to fragment the ICMP datagram on sending. Setting the DF bit on the following replies would help to maintain the PMTU between the two systems, if for any reason, the PMTU would be decreased. For example, because the datagram have used another route to the destined system.

Sending immediately another ICMP Query message type to this particular HP-UX 11.x operating system based machine, will not result in the PMTU discovery process to be repeated. The DF Bit would be set within the ICMP Query reply. Expect a threshold to be maintained by the HP-UX 11.x. When reached the next time we query this host with any type of communication, the process of determining the PMTU using ICMP Echo request will begin again.

Why this method is bound to failure?

- Some ISPs would configure their routers not to allow fragmented ICMP datagrams through. I have encountered this behavior with different ISPs I have used.
- Some machines would be configured not to reply for an ICMP Echo requests coming from the Internet (if you read all of this research paper you'll do that).
- This ability can be used for a denial-of-service attack with the HP-UX 10.30, and/or 11.0x machines used as an amplifier for these attacks. Infact, HP has released a security bulletin dated February 13, 2000 about some issues regarding this PMTU discovery capability with ICMP. The bulletin states that "Depending upon the amount and nature of the inbound traffic, an HP-UX 10.30/11.00/11.04 system can be used to flood a target system with IP packets which could result in a denial of service"<sup>30</sup>.
- Easy identification of HP-UX 10.30, 11.0x machines that had the default behavior not changed.

This gives us the ability to distinguish between Sun Solaris machines, HP-UX 11.0x/10.30 machines, and AIX 4.3.x based machines.

Sun Solaris sets the DF bit with the ICMP Query replies the operating system answers for, in order to support its global PMTU discovery process. If the networking link will not let the ICMP Query reply to get back to the querying host, because the MTU used is higher than the allowed

<sup>30</sup> HP Security Bulletin - "Security Vulnerability with PMTU strategy (revised)", February 13, 2000.